1302

Instruction Manual

Vol. 2

Interface Manual

Multi-gas Monitor Type 1302

This volume provides users with detailed information about how to operate the IEEE–488 / RS–232–C interface of the Multi-gas Monitor Type 1302.

Brüel & Kjær

SAFETY CONSIDERATIONS

PLEASE READ THESE SAFETY CONSIDERATIONS CAREFULLY AND MAKE SURE YOU UNDERSTAND THEM PROPERLY BEFORE YOU START OPERATING THE MULTI-GAS MONITOR TYPE 1302

EXPLOSION HAZARD

⚠ **THE MULTI–GAS MONITOR TYPE 1302 IS NOT DESIGNED FOR USE IN POTENTIALLY EXPLOSIVE ENVIRONMENTS.**

This means that the instrument must **not** be placed and operated in an area with a potentially explosive atmosphere.

When monitoring potentially flammable or toxic gases it is essential that:

1) the instrument itself is placed in a well-ventilated area **outside** the potentially hazardous zone; and
2) that a sufficiently long tube is connected to the Monitor's "Air Outlet" so that the measured gas is carried **away** to the open air or to an extraction and/or filtration unit.

TO AVOID THE POSSIBILITY OF AN EXPLOSION, MONITORING OF FLAMMABLE GASES **IN EXPLOSIVE CONCENTRATIONS** MUST NEVER BE ATTEMPTED

⚠ **AVOID WATER CONDENSATION IN THE ANALYSIS CELL.**

Liquids must be prevented from entering the analysis cell. It is therefore important that warm humid gases are not drawn into a cold analysis cell because condensation will take place. If such a situation is likely to occur you should ensure that the gases are drawn through a water-trap filter before they enter the air-inlet of the 1302. This will condense out water vapour in the gases and thus prevent condensation within the analysis cell. The water-trap filter should be used in the immediate environment of the Monitor so it maintains either the same temperature, or a lower temperature, than the Monitor.

Note: that some gases may be absorbed by the water trapped in the filter. This will reduce the gases' concentration.

MULTI–GAS MONITOR
TYPE 1302

From serial number 1487212

March 1991

SAFETY CONSIDERATIONS

The Multi-gas Monitor Type 1302 has been designed and tested in accordance with IEC Publication 348, *Safety Requirements for Class 1 Electronic Measuring Apparatus*, and has been supplied in safe condition. This Instruction Manual contains information and warnings which must be followed by the user to ensure safe operation and to retain the Multi-gas Monitor in safe condition. Special note should be made of the following:

APPLICATION OF POWER

Before using the Multi-gas Monitor check that its Mains Voltage selector is set to available mains voltage supply and that the correct fuse is installed (details are provided in Chapter 2 of Volume 1).

SAFETY SYMBOLS

⚠ The apparatus is marked with this symbol when it is important that the user refers to the associated warning statements given in the Instruction Manual.

⊥ Chassis terminal ⏚ Safety earth terminal ↯ Hazardous voltage.

EXPLOSION HAZARD

The Multi-gas Monitor is not designed to be used in potentially explosive environments. It should not be operated in the presence of flammable liquids or gases

The Monitor contains a lithium battery. Under no circumstances should this battery be removed or replaced by the user as there is a danger of explosion. The lithium battery should only be removed by a Brüel & Kjær representative.

WARNINGS

Switch off all power to instruments before either connecting or disconnecting interface cables.

Whenever it is likely that the correct function or operating safety of the Multi-gas Monitor has been impaired, it must be made inoperative and be secured against unintended operation.

Any adjustment, maintenance and repair of the open Multi-gas Monitor Type 1302 while it is still connected to an AC mains voltage supply must be avoided as far as possible, and, if inevitable, must only be performed by a qualified person who is aware of the hazard involved.

CONTENTS

1. INTRODUCTION **1**

1.1. PURPOSE OF THIS MANUAL 1
1.2. SUMMARY OF CONTENTS 1

2. THE IEEE–488 INTERFACE **2**

2.1. INTRODUCTION 2
Data Conventions 2
2.2. IEEE–488 FUNCTIONS IMPLEMENTED 2
Local Lockout and Remote Enable 3
2.3. ADDRESSING THE TYPE 1302 4
Setting the Address 5
2.4. MESSAGE SYNTAX 5

3. THE RS–232–C INTERFACE **6**

3.1. INTRODUCTION 6
Interface Connector 6
Data Lines 7
Control Lines 7
3.2. INTERCONNECTION 7
3.3. INTERFACE SET-UP 9
Baud Rate 9
Parity 9
Stop Bit 9
Data Bit 9
Handshake Method 10
Hardwire Mode 10
3.4. TRANSMISSION INTERRUPT 10

4. THE INTERFACE MESSAGE **11**

4.1. MESSAGE FORMAT 12
4.2. THE INDIVIDUAL MESSAGES 14
Accept_Message 14
Alarm_Start 14
Alarm_Stop 15
Beep 15
Binary_Numeric_Order 15
Calibration_Data_Binary 16
Channel_Lockout 17
Click 18

Control_SRQ 18
Current_Filter 21
Current_Setup 21
Define_Terminator 22
Delete_Measurement 22
Display_State 23
Error_Stop 24
Execute_State? 25
Extended_Control 26
External_Pump 26
Identification? 27
Identify? 28
Keyboard_Lockout 28
Measurement_State? 29
Output_Calibration_Data? 29
Output_Error_Dump? 30
Output_Header 30
Output_Measurement? 31
Output_Setup_Data? 31
Output_Special_Compilation? 32
Pass_Control_Back 33
Plot_Measurement 33
Pop_Up_Display_Buffer? 33
Print_Calibration_Data 34
Print_Error_Dump 34
Print_Measurement 35
Print_Setup_Data 35
Print_Special_Compilation 36
Push_Key 36
Recall_Measurement 38
Relay 38
Remote_Display_Buffer 39
Reset_Command 40
Reset_Status_Byte 40
Reset_System 40
Self_Test? 41
Service_Request_Enable 42
Setup 43
Software_Identification? 46
Start_Delayed_Measurement 47
Start_Measurement 47
Status_Byte? 47
Stop_Measurement 48
Store_Measurement 48
Synchronize 49
System_Display_Buffer? 50
Time_Since_Reset? 50
4.3. THE IEEE–488 BUS MESSAGES 50
IEEE_Attention 51
IEEE_Clear 51
IEEE_Enter? 52
IEEE_Interface_Clear 52
IEEE_Local 52
IEEE_Local_Lockout 53
IEEE_Output 53
IEEE_Parallel_Poll? 54

IEEE_PPC 54
IEEE_PPU 55
IEEE_Remote 55
IEEE_Serial_Poll? 55

5. APPENDIX 56

5.1. INTERFACE CHARACTER DATA 56
5.2. PROGRAMMING EXAMPLE 56
5.3. ERROR MESSAGES 61
5.4. INTRODUCTION TO SERVICE REQUESTS 64
The Status Byte 65
The Service Request Enable Register 65
The Event and Event Enable Registers 65

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_10314

1. INTRODUCTION

1.1. PURPOSE OF THIS MANUAL

This manual describes the programming and operation of the serial and parallel interfaces of the Type 1302 Gas Monitor. It is assumed that users are familiar with manual operation of the gas monitor, as described in Vol.1 of the Instruction Manual (Brüel & Kjær publication BE 1066–11).

1.2. SUMMARY OF CONTENTS

Chapters 2 and 3 give a brief description of the gas monitor IEEE–488/RS–232–C interface. Only the most important specifications are described. Chapter 2 also deals with the syntax of the interface message.

Chapter 4 describes the individual messages.

The interface character data with their corresponding minimum codes and the error messages which can occur during data transfer are listed in Chapter 5.

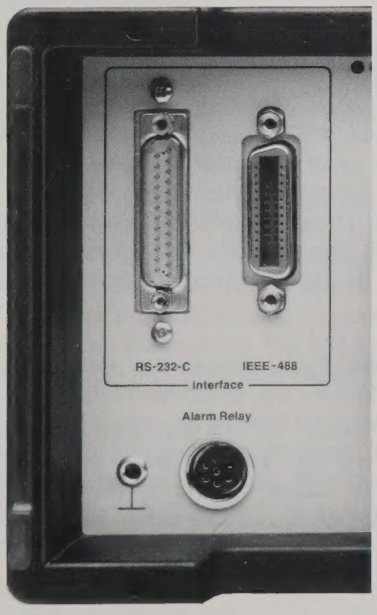


Fig. 1.1. The interface connections on the rear panel of the Type 1302

WARNING! Switch off all equipment before connecting or disconnecting the digital interface. Failure to do so could damage the equipment.

2. THE IEEE-488 INTERFACE

2.1. INTRODUCTION

The digital interface of the gas monitor is designed according to ANSI/IEEE Std 488–1978, "IEEE Standard Digital Interface for Programmable Instrumentation". Since the only significant difference between this and IEC Publication 625–1 is in the type of connector specified, compatibility with these standards is only a question of which connecting cable is used.

The gas monitor is connected to other instruments with IEEE/IEC interfaces using the following Brüel & Kjær cables and connectors.

1. Cable AO 0265, which has an IEEE–488 standard connector at each end, connects the gas monitor to other instruments equipped with the IEEE–488 connector.
2. Cable AO 0264, with the IEEE–488 connector at one end and an IEC–625–1 connector at the other end, connects the gas monitor to instruments equipped with the IEC–625–1 connector.

2.1.1. Data Conventions

The codes and formats used by the gas monitor in sending and receiving data and/or messages via the IEEE/IEC interface have been designed according to the recommendations of IEEE Std 728–1982, "IEEE Recommended Practice for Code and Format Conventions (For Use with ANSI/IEEE Std 488–1978)". This standard is identified by the IEEE as a recommended practice rather than an absolute standard. It seeks to define standard forms for the transfer of various types of data on the IEEE–488 interface, and so to improve the level of communication between instruments equipped with the interface. In particular, IEEE 728 recommends that all messages use ASCII/ISO 7-bit coding and it defines standard header messages, data formats and message separators and terminators.

It is important to note that although the gas monitor interface is designed according to IEEE Std 488 and follows the recommendations of IEEE Std 728, absolute compatibility with IEC or IEEE/ANSI interfaces designed by other manufacturers cannot be unconditionally guaranteed, since differences can occur within the limits of the specifications. Any problems encountered, however, will be of a software rather than a hardware nature. Where compatibility is in doubt, contact your local Brüel & Kjær representative for further information.

2.2. IEEE–488 FUNCTIONS IMPLEMENTED

The interface of the gas monitor implements the following functions as specified by the IEEE–488 standard. The Sections referred to are the relevant sections of the IEEE Std 488–1978 which specify the functions. The equivalent Clauses of IEC 625–1 are given in parentheses.

Section	2,3	Source Handshake (SH) Interface Function, (Clause 6) SH 1 — complete capability
Section	2,4	Acceptor Handshake (AH) Interface Function, (Clause 7) AH 1 — complete capability
Section	2,5	Talker (T) Interface Function, (Clause 8) T 5 — complete capability
Section	2,6	Listener (L) Interface Function, (Clause 9) L 3 — complete capability
Section	2,7	Service Request (SR) Interface Function, (Clause 10) SR 1 — complete capability
Section	2,8	Remote Local (RL) Interface Function, (Clause 11) RL 1 — complete capability
Section	2,9	Parallel Poll (PP) Interface Function, (Clause 12) PP 1 — remote configuration
Section	2,10	Device Clear (DC) Interface Function, (Clause 13) DC 1 — complete capability
Section	2,12	Controller (C) Interface Function, (Clause 15) C 1,2,3,4,5 — complete capability
		All other functions — no capability

For further details of the above functions, refer to the relevant sections of the IEEE or IEC standards.

2.2.1. Local Lockout and Remote Enable

The IEEE–488 interface Remote ENable (REN) line allows selection between manual operation via the front panel controls of the gas monitor and remote operation via the interface bus. The REN line is controlled by the device currently specified as system controller.

If the gas monitor is addressed after the REN line has been set true, it goes into the "Remote" state and the **"Remote"** LED at the bottom right–hand corner of the front panel is lit. The "Remote" state is only maintained while the REN line is true. While in the remote state, none of the pushkeys are active except the **"Stand By"**, which is the gas monitor on/off key, and the **"Local"** key. Pressing the **"Local"** key will return the gas monitor to its "Local" state, deactivating the **"Remote"** LED and returning control to the front panel, although the REN line will still be true.

The system controller can also disable the **"Local"** pushkey by issuing the LLO **(Local Lockout)** command. In this case, the gas monitor remains under remote (i.e. interface bus) control until the system controller issues the GTL **(Go To Local)** command. The **"Stand By"** key can still be used. Both LLO and GTL are interface bus commands, as defined in the interface standards.

2.3. ADDRESSING THE TYPE 1302

Each device in an IEEE–488 interface bus system has at least one listener and / or talker address depending on its function. When an interface controller contacts a device over the IEEE–488 interface, it sends a device address, which will contain the appropriate talker or listener address. This address is in ISO 7-bit code (or equivalent ASCII), bit 8 being unused for addressing and therefore remaining at logic 0. The same setting of the first 5 bits is used regardless of whether the gas monitor is addressed as a talker or a listener. This is because only the settings of bit 6 and bit 7 determine whether the gas monitor is being addressed as a listener or a talker. The default setting of the first 5 bits is the binary equivalent of decimal 14. Table 2.1 gives a list of the possible addresses with the corresponding ASCII characters for the different address modes. All values in Table 2.1 are legal addresses for the gas monitor.

Address Code (decimal)	Address Code (Binary)					ASCII Address Characters	
	A5	A4	A3	A2	A1	Listen	Talk
0	0	0	0	0	0	SP	@
1	0	0	0	0	1	!	A
2	0	0	0	1	0	"	B
3	0	0	0	1	1	#	C
4	0	0	1	0	0	$	D
5	0	0	1	0	1	%	E
6	0	0	1	1	0	&	F
7	0	0	1	1	1	'	G
8	0	1	0	0	0	(	H
9	0	1	0	0	1	)	I
10	0	1	0	1	0	*	J
11	0	1	0	1	1	+	K
12	0	1	1	0	0	,	L
13	0	1	1	0	1	–	M
14	0	1	1	1	0	.	N
15	0	1	1	1	1	/	O
16	1	0	0	0	0	0	P
17	1	0	0	0	1	1	Q
18	1	0	0	1	0	2	R
19	1	0	0	1	1	3	S
20	1	0	1	0	0	4	T
21	1	0	1	0	1	5	U
22	1	0	1	1	0	6	V
23	1	0	1	1	1	7	W
24	1	1	0	0	0	8	X
25	1	1	0	0	1	9	Y
26	1	1	0	1	0	:	Z
27	1	1	0	1	1	;	[
28	1	1	1	0	0	<	\
29	1	1	1	0	1	=	]
30	1	1	1	1	0	>	^

T02618GB0

Table 2.1. Listener/Talker addresses (shaded area indicates the Type 1302 default value)

2.3.1. Setting the Address

The listen and talk addresses over which the gas monitor can be contacted via their IEEE–488 interface are selected through the "set-up" tree on the front panel (see Vol. 1 section 7.5.2). The selections are : SET-UP - CONFIGURATION - SYSTEM - COMMUNICATION - IEEE-488.

WARNING! You must partially reset the Type 1302 if the IEEE configuration is changed, for example if the device address is changed. See section 7.5 of Vol. 1 of the Instruction Manual.

2.4. MESSAGE SYNTAX

A general diagram of the Message syntax is given in Fig. 2.1. The "Message Header" describes the task to be performed. This consists of one word, or two or more words separated by a concatenator (_). A hyphen (–) or a full stop (.) are also accepted as word separators. The words may be written in full, or shortened to a minimum code, as long as the first letter(s) of each word corresponds to the "Minimum Code" in Table 4.1 (a) and (b).

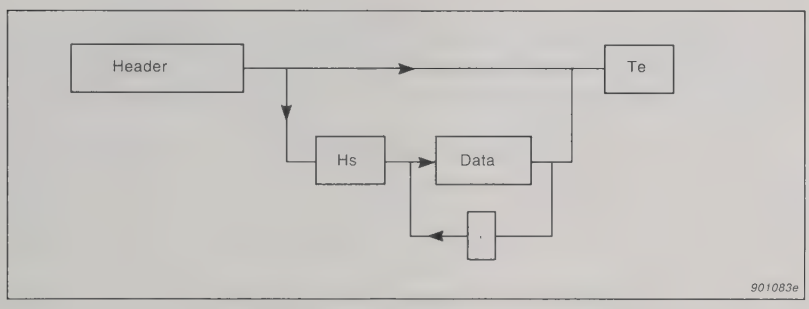


Fig. 2.1. Message syntax diagram

Notes

Messages where the header includes a "?" are output messages, i.e. the gas monitor will respond by reading out the relevant data when addressed as a talker.

Example

OUTPUT_SPECIAL_COMPILATION? or
OUT_SPEC_COMP? or
O_SP_C?

will give the same result.

The "Data" (or parameters), if present, are used to further specify the Message. Parameters consisting of character data, have the same syntax as the message header. To separate the "Message Header" from the Data, a "Header Separator" character is used. Data is separated by a comma (,). Individual Messages can be sent together separated by a semicolon (;). The Message specification ends with a "Message Terminator" character.

3. THE RS-232-C INTERFACE

3.1. INTRODUCTION

The interface of the Type 1302 conforms with the EIA standard RS–232–C, which is equivalent to the CCITT V.24 recommendation.

The interface is coupled as "Data Terminal Equipment" (DTE), and it operates in full duplex mode, which means that the interface is capable of operating in both directions simultaneously.

3.1.1. Interface Connector

The interface connector is a 25-pin D-range male connector. It is located on the rear panel of the monitor as shown in Fig. 1.1. Fig. 3.1 shows the numbering of the pins. Pin definitions are given in Table 3.1.

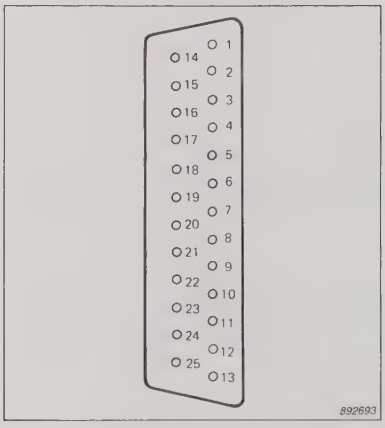


Fig. 3.1. The interface connector of the Type 1302

Pin No.	RS–232–C	CCITT V.24	Description	Mnemomic	Direction
1	AA	101	Protective Ground	P GND	—
2	BA	103	Transmitted Data	TxD	From 1302
3	BB	104	Received Data	RxD	To 1302
4	CA	105	Request to Send	RTS	From 1302
5	CB	106	Clear to Send	CTS	To 1302
6	CC	107	Data Set Ready	DSR	To 1302
7	AB	102	Signal Ground	S GND	—
8	CF	109	Data Carrier Detect	DCD	To 1302
20	CD	108.2	Data Terminal Ready	DTR	From 1302

T02664GB0

Table 3.1. Pin definitions of the RS–232–C interface

3.1.2. Data Lines

Pins 2 and 3, Transmitted Data and Received Data, are data lines. For data lines, RS–232–C specifies that:

- A voltage below –3 V signifies a binary 1
- A voltage above +3 V signifies a binary 0
- When a data line is passive, it is held in the binary 1 condition.

Data transmission is asynchronous as shown in Fig. 3.2.

Most of the data is transmitted in the form of ASCII codes. All ASCII codes given in the following text are decimal numbers.

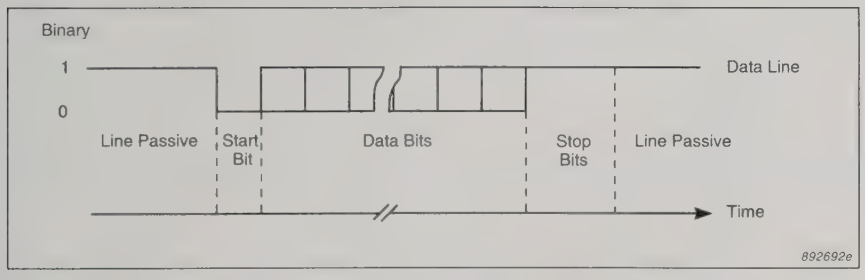


Fig. 3.2. Asynchronous transmission of character data

3.1.3. Control Lines

Pins 4, 5, 6, 8 and 20 are control lines. For control lines, RS–232–C specifies that:

- A voltage above +3 V signifies the "on" state
- A voltage below –3 V signifies the "off" state.

The use of the control lines is described in section 3.3.

3.2. INTERCONNECTION

A null-modem cable must be used to interface the monitor to another DTE-coupled device (computer or printer), so that the DTEs appear to each other to behave like modems. Examples of null-modem cross connections which may be used with the monitor are given in Fig. 3.3 and Fig. 3.4. The cables for interconnection can be obtained from Brüel & Kjær, as listed in Table 3.2. below, or else they can be constructed.

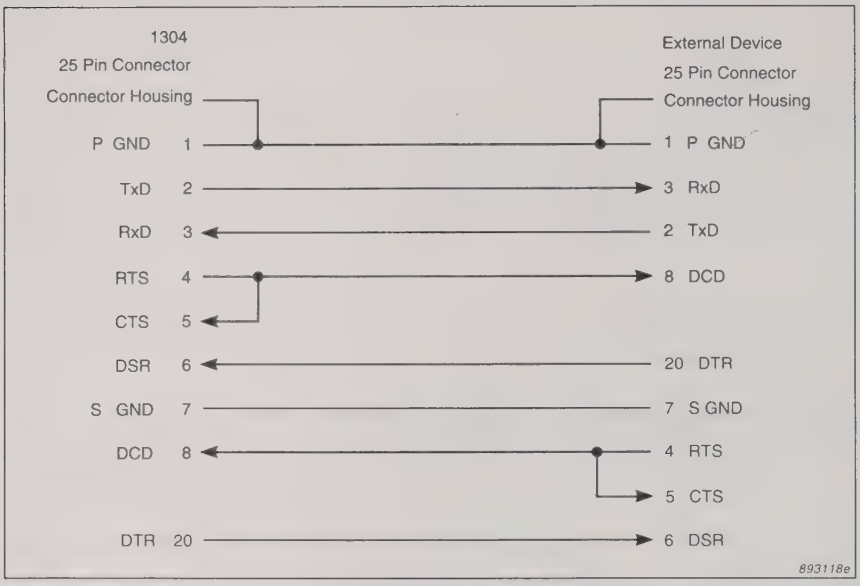


Fig. 3.3. Null-modem cable with 25-pin connectors at both ends (Brüel & Kjær order numbers WL0946 and WL0947)

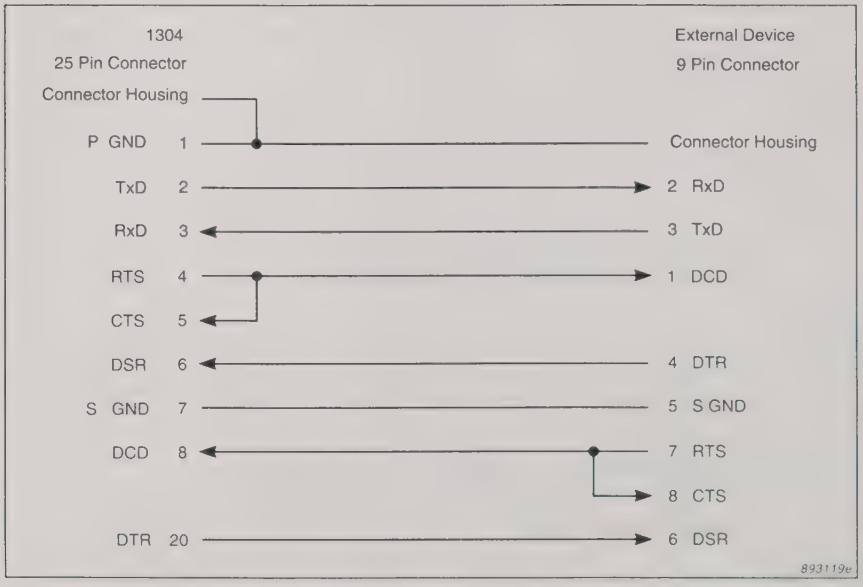


Fig. 3.4. Null-modem cable with one 25-pin connector and one 9-pin connector (Brüel & Kjær order number WL0945)

Order No.	Connectors	Used to Interface to
WL 0945	25 pin female 9 pin female	Computer or printer with 9 pin male RS–232–C connector
WL 0946	25 pin female 25 pin female	Computer or printer with 25 pin male RS–232–C connector
WL 0947	25 pin female 25 pin male	Computer or printer with 25 pin female RS–232–C connector

T02416GB0

Table 3.2. Brüel & Kjær RS–232–C null modem cables

3.3. INTERFACE SET-UP

The "set-up" tree on the front panel enables the user to specify the interface set-up of the monitor (see Instruction Manual Vol. 1 section 7.5.1). The selections are : SET-UP - CONFIGURATION - SYSTEM - COMMUNICATION - RS-232-C.

WARNING! You must partially reset the Type 1302 if the RS–232–C configuration is changed, for example if the device address is changed. See section 7.5 of Vol. 1 of the Instruction Manual.

3.3.1. Baud Rate

The baud rate is the speed of data transmission. Possible baud rates are 300, 600, 1200, 2400, 4800 and 9600 bits/sec.

3.3.2. Parity

Checking the parity of a transmitted byte is a way of detecting transmission errors.

Parity is a system which consists of selectively adding a "1" bit to bit patterns in order to cause the bit patterns to have either an odd number of bits (odd parity) or an even number of bits (even parity).

Receiving hardware checks the number of "1" bits in each pattern and indicates an error condition if a pattern with an even number of "1" bits is found in an odd parity system, or an odd number in an even parity system. The possible options are therefore:

- Even
- Odd
- None.

Where "None" is selected, no parity check is performed.

3.3.3. Stop Bit

The stop bit is used to terminate a transmitted character (see Fig. 3.2). Together with the start bit, the stop bit frames the data bits and is used to synchronise data transmission. The monitor can use either one or two stop bits.

3.3.4. Data Bit

Setting the number of data bits determines how many bits of data will be contained in each transmitted character. Possible options are seven or eight data bits.

3.3.5. Handshake Method

The handshake method is the method used by the monitor and the external device to synchronize their communication. There are three possible options:

- X-On/X-Off handshake
- Hardwired handshake
- None.

When X-On/X-Off handshake is used, the monitor transmits the X-Off character (ASCII 19) when it can handle no more input data. When the monitor is once again ready to receive data, it transmits the X-On character (ASCII 17) over the interface. Similarly, when the monitor receives an X-Off character during a transmission, it stops transmitting until it receives the X-On character. If the X-On character does not arrive within 30 seconds, the monitor regards the transmission as aborted and issues a time-out warning.

In hardwired-handshake mode, the monitor sets the DTR control line (pin 20) off when it will accept no more data. When it is ready to receive more data, the DTR line is set on. When the external device sets the DSR control line (pin 6 of the monitor) off, the monitor stops transmitting until DSR goes back on. If the DSR line remains off for more than 30 seconds, the monitor issues a time-out warning.

If "None" is selected, handshaking is disabled.

If either "Hardwired" or "None" is selected and the external device attempts to use X-On/X-Off handshaking, the monitor displays a warning. Similarly, if "X-On/X-Off" or "None" is selected and DSR is set off by the external device, a warning will appear.

3.3.6. Hardwire Mode

This parameter determines how the monitor uses the control lines of the interface. Three set-ups exist:

- 3–Wire mode
- Switched–Line mode
- Leased–Line mode.

In 3–wire mode, only the data lines are used. All control lines are ignored.

This means that X-On/X-Off handshake is the only handshake method available.

In switched–line mode, the CTS and DSR control lines are monitored. If one of the two control lines is set off, data transmission from the monitor is blocked. It is recommended that the external device uses the DSR line for handshaking.

In leased–line mode, the CTS, DSR and DCD control lines are monitored. CTS and DSR have the same effect as in switched line mode. If the DCD input control line goes off, the monitor ignores all incoming data.

3.4. TRANSMISSION INTERRUPT

If the monitor receives a Ctrl-C (ASCII 3) character over the interface during transmission of data, the transmission will be aborted.

4. THE INTERFACE MESSAGE

For remote control of the Type 1302 over the IEEE–488/RS–232–C interface bus, various interface messages can be specified, a list of which is given in Table 4.1 (a) and (b).

Alphabetical list of interface messages					
	Minimum code	Input Job	Output Job	RS–232–C	IEEE 488
ACCEPT_MESSAGE	A_M	I	0	R	I
ALARM_START	A_STA	I		R	I
ALARM_STOP	A_STO	I		R	I
BEEP	B	I		R	I
BINARY_NUMERIC_ORDER	B_N_O	I		R	I
CALIBRATION_DATA_BINARY	C_D_B	I	0	R	I
CHANNEL_LOCKOUT	C_L	I	0	R	I
CLICK	C	I		R	I
CONTROL_SRQ	CO_SRQ	I	0	R	I
CURRENT_FILTER	C_F	I	0	R	I
CURRENT_SETUP	CU_SE	I	0	R	I
DEFINE_TERMINATOR	D_T	I			I
DELETE_MEASUREMENT	D_M	I		R	I
DISPLAY_STATE	D_S	I	0	R	I
ERROR_STOP	ER_S	I		R	I
EXECUTE_STATE?	EX_S?		0	R	I
EXTENDED_CONTROL	E_C	I		R	I
EXTERNAL_PUMP	E_P	I	0	R	I
IDENTIFICATION?	*IDN?		0	R	I
IDENTIFY?	ID?		0	R	I
IEEE_ATTENTION	I_A	I		R	
IEEE_CLEAR	I_C	I		R	
IEEE_ENTER?	I_E?		0	R	
IEEE_INTERFACE_CLEAR	I_I_F_C	I		R	
IEEE_LOCAL	I_L	I		R	
IEEE_LOCAL_LOCKOUT	I_L_L_O	I		R	
IEEE_OUTPUT	I_O	I		R	
IEEE_PARALLEL_POLL?	I_P_P?		0	R	
IEEE_PPC	I_P_P_C	I		R	
IEEE_PPU	I_P_P_U	I		R	
IEEE_REMOTE	I_R	I		R	
IEEE_SERIAL_POLL	I_S_P		0	R	
KEYBOARD_LOCKOUT	K_L	I	0	R	I
MEASUREMENT_STATE?	M_S		0	R	I
OUTPUT_CALIBRATION_DATA?	O_C_D?		0	R	I
OUTPUT_ERROR_DUMP?	O_E_D?		0	R	I
OUTPUT_HEADER	O_H	I		R	I
OUTPUT_MEASUREMENT?	O_M?		0	R	I
OUTPUT_SETUP_DATA?	O_SE_D?		0	R	I
OUTPUT_SPECIAL_COMPILE?	O_SP_C?		0	R	I

Table 4.1(a). List of interface messages

Alphabetical list of interface messages (continued)					
	Minimum code	Input Job	Output Job	RS–232–C	IEEE 488
PASS_CONTROL_BACK	*PCB	I		R	I
PLOT_MEASUREMENT	PL_M	I		R	I
POP_UP_DISPLAY_BUFFER?	P_U_D_B?		O	R	I
PRINT_CALIBRATION_DATA	P_C_D	I		R	I
PRINT_ERROR_DUMP	P_E_D	I		R	I
PRINT_MEASUREMENT	PR_M	I		R	I
PRINT_SETUP_DATA	P_SE_D	I		R	I
PRINT_SPECIAL_COMPILE.	P_SP_C	I		R	I
PUSH_KEY	P_K	I	O	R	I
RECALL_MEASUREMENT	R_M	I		R	I
RELAY	R	I	O	R	I
REMOTE_DISPLAY_BUFFER	R_D_B	I	O	R	I
RESET_COMMAND	*RST	I		R	I
RESET_STATUS_BYTE	R_S_B	I		R	I
RESET_SYSTEM	RESET_SY	I		R	I
SELF_TEST?	*TST?		O	R	I
SERVICE_REQUEST_ENABLE	S_R_E	I	O	R	I
SETUP	SE	I	O	R	I
SOFTWARE_IDENTIFICATION?	S_I?		O	R	I
START_DELAYED_MEASUREMENT	S_D_M	I		R	I
START_MEASUREMENT	STA_M	I		R	I
STATUS_BYTE?	*STB?		O	R	I
STOP_MEASUREMENT	STOP_M	I		R	I
STORE_MEASUREMENT	STOR_M	I		R	I
SYNCHRONIZE	SY?	I	O	R	I
SYSTEM_DISPLAY_BUFFER?	SY_D_B?		O	R	I
TIME_SINCE_RESET?	T_S_R		O	R	I

T02605GB0

Table 4.1(b). List of interface messages

4.1. MESSAGE FORMAT

The same message format is used for programming both the IEEE–488 and RS–232–C interfaces. Differences in individual messages are described in the relevant section for that message, e.g. **C**hannel_**L**ockout (section 4.2.7).

These standards will be used in the description of messages:

- [Hs]

  This is a "Header Separator". It must be inserted between the header and the first data field. A <SPACE> is always used as the header separator.

- [Te]

  This is a "Message Terminator". With the monitor the default terminator is <LF> (i.e. <CTRL>+<J>). It can be changed via the IEEE–488 interface but not via the RS–232–C interface. The change can be made using the message **D**efine_**T**erminator. (See 4.2.12)

Note: When programming the monitor via the interface, it sends data as a series of one or more text lines. Each of these lines is terminated by a "text line terminator". This may be changed by the user within the "set-up" tree. There are three possibilities; either <LF>, or <CR>, or <CR><LF>. When the monitor has finished sending data, i.e. after all text lines have been sent, a final terminator is sent. This is the same as the "Message Terminator" described above. A single text line sent by the monitor in reply to an interface message might be, for example;

"B&K 1302 <CR><LF><LF>"

In this case, <CR><LF> has been selected as the text line terminator, and, because only one text line is sent, another <LF> is sent immediately following this indicating that the monitor has finished sending data. Interface programs reading data from the monitor should take account of this.

- BLOCK DATA

  This field describes the way in which block data is sent over the bus by the monitor. The block data fields conform to the general block data format as defined by ANSI/IEEE Standard 488.2. It takes the form:

  #C n..n [n..n data bytes]

  Length count C is an integer between 1 and 9 inclusive (a length count of 0; Block 0 data, does not occur with the Type 1302). Length count C gives the number of characters in the following block length field, n..n. This field is between one and nine characters long and gives the number of data bytes found after it. For example:

  #4 1200 [1200 data bytes]

- NRx

  These are numbers according to IEEE–488 standards, where x may be 1, 2, or 3. NR1 is always an integer, NR2 is real without an exponential, and NR3 is real with an exponential.

- Anything written in closed boxes (apart from the above) is the minimum code necessary for executing the message correctly.

- Anything written in broken-line boxes will be described in detail in the text.

- /***** PROTECTED xx *****/

  This means that the message will be recognized by the monitor only if it has been unlocked. The "unlocking" can be done with the message **E**xtended_**C**ontrol, see section 4.2.17.

  Note:

  You must "unlock" the message each time it is sent.

Message Note

Messages can be input using upper or lower-case characters.

4.2. THE INDIVIDUAL MESSAGES

4.2.1. ACCEPT_MESSAGE

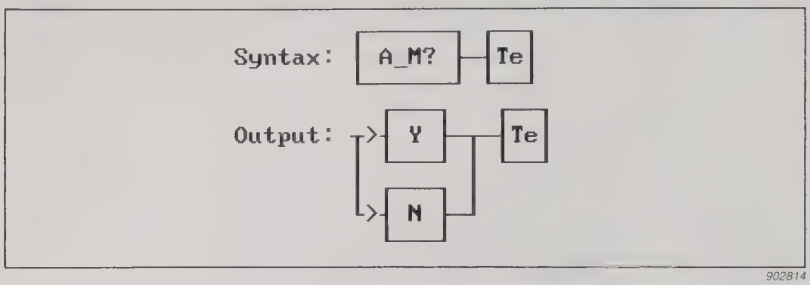


Fig. 4.1. Syntax for Accept_Message?

Indicates if there is a message to acknowledge.

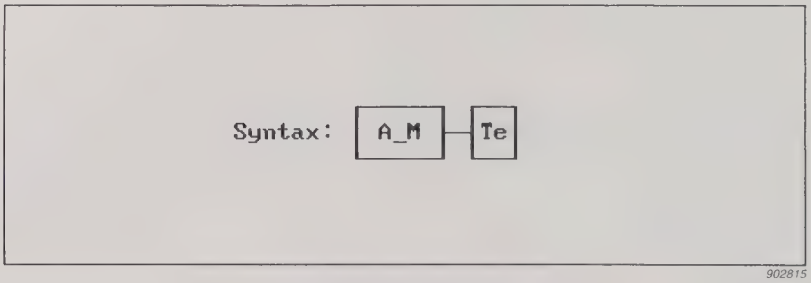


Fig. 4.2. Syntax for Accept_Message

Acknowledges the message. Corresponds to pressing the "**Status**" key on the monitor front panel, if there is a message.

4.2.2. ALARM_START

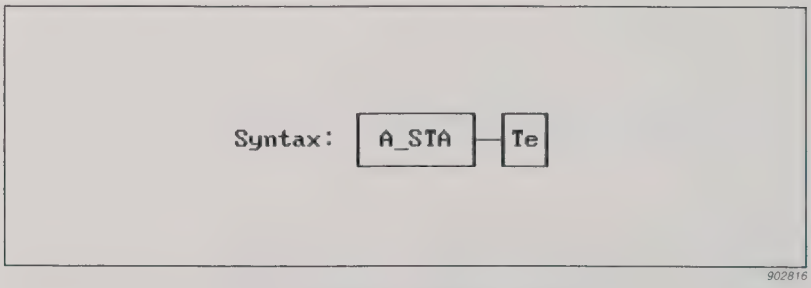


Fig. 4.3. Syntax for Alarm_STArt

Activates the acoustic alarm of the monitor.

4.2.3. **A**LARM_**STO**P

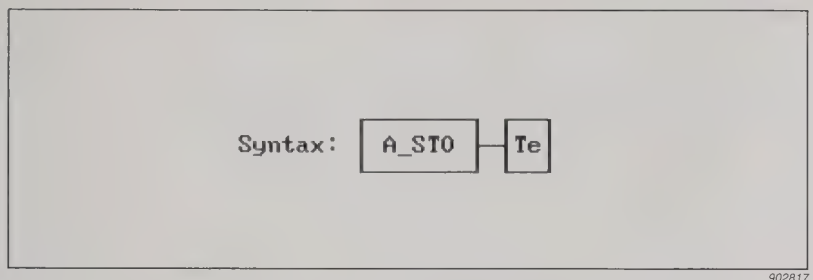


Fig. 4.4. Syntax for Alarm_STOp

Stops the acoustic alarm of the monitor.

4.2.4. **B**EEP

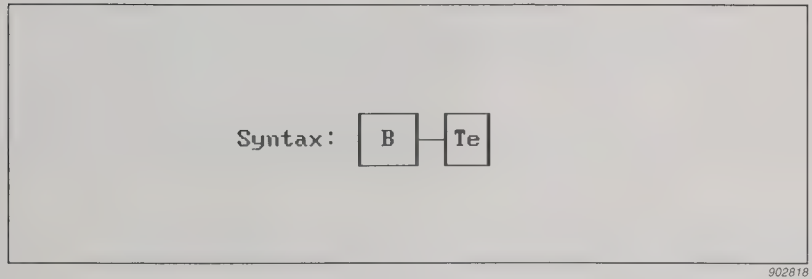


Fig. 4.5. Syntax for Beep

The monitor emits a tone.

4.2.5. **B**INARY_**N**UMERIC_**O**RDER

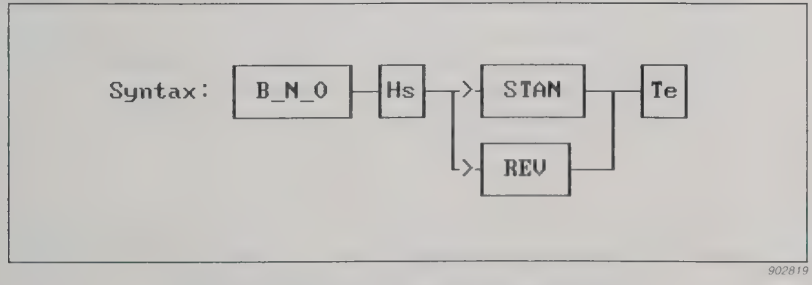


Fig. 4.6. Syntax for Binary_Numeric_Order

Some interface messages cause the monitor to give output in binary form and the order in which the bytes are given may be determined by using **B_N_O.** Typing STAN or REV as shown in the syntax diagram will result in the following byte order:

STANdard	Most significant byte first
REVerse	Least significant byte first

For example, the number 300 read out as a 16 bit word in STANdard order will be:

00000001 00101100

The least significant byte will be 44 and the most significant byte will be 1 (corresponding to 256 when the 2 bytes are taken as a 16 bit word).

The number 300 read out in REVerse order will be:

00101100 00000001

4.2.6. **C**ALIBRATION_**D**ATA_**B**INARY

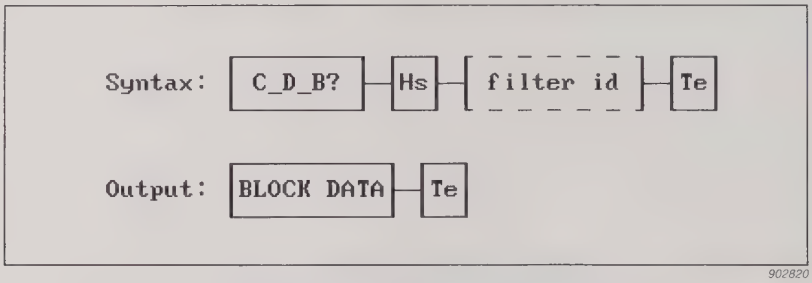


Fig. 4.7. Syntax for Calibration_Data_Binary?

Reads out calibration data in binary form.
This message is primarily used as a backup for calibration data.
"filter id" is one of the following:

ALl	Data for all filters
Filter_**A**	Data for filter A
Filter_**B**	Data for filter B
Filter_**C**	Data for filter C
Filter_**D**	Data for filter D
Filter_**E**	Data for filter E
Filter_**W**	Data for filter W

/***** PROTECTED 15 *****/

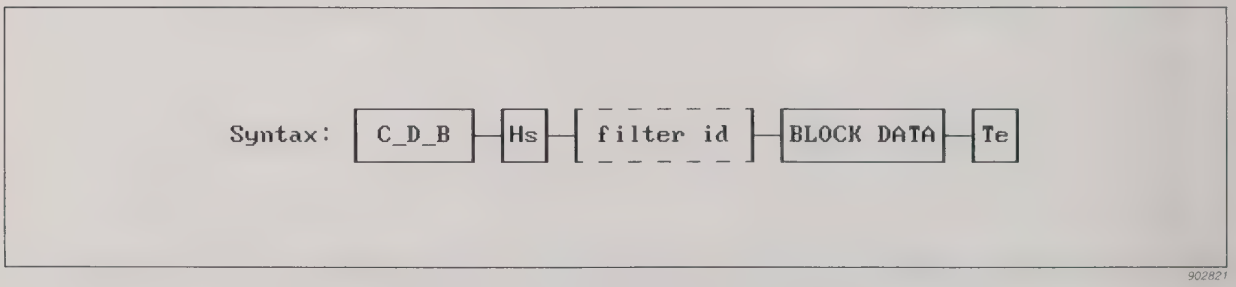


Fig. 4.8. Syntax for Calibration_Data_Binary

Input of calibration data. This may also be used for back-up of the calibration.

4.2.7. **C**HANNEL_**L**OCKOUT

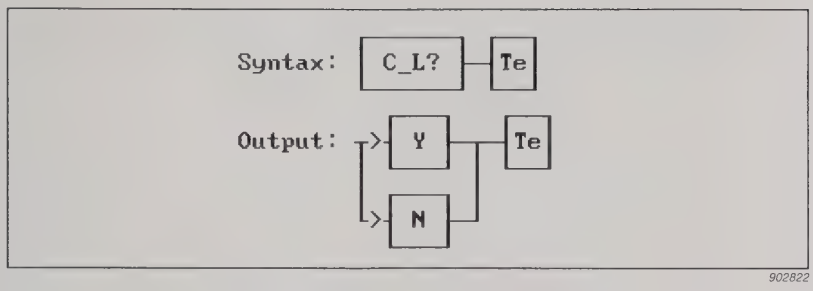


Fig. 4.9. Syntax for Channel_Lockout?

If the message is used via IEEE–488, the readout from the monitor will indicate whether the RS–232–C interface is blocked for messages. If the message is used via RS–232–C, the readout from the monitor will indicate whether the IEEE–488 interface is blocked for messages.

Yes The IEEE–488/RS–232–C interface is locked out.
No The IEEE–488/RS–232–C interface is open.

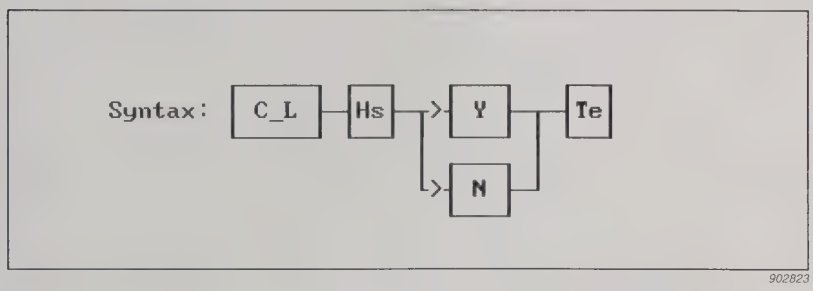


Fig. 4.10. Syntax for Channel_Lockout

If the message is used via IEEE–488, the RS–232–C interface is blocked/opened for messages. If the message is used via RS–232–C, the IEEE–488 interface is blocked/opened for messages.

4.2.8. **C**LICK

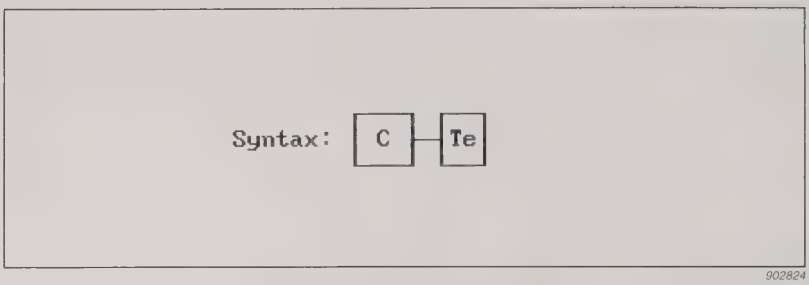


Fig. 4.11. Syntax for Click

The monitor emits a "clicking" sound.

4.2.9. **CO**NTROL_**SRQ**

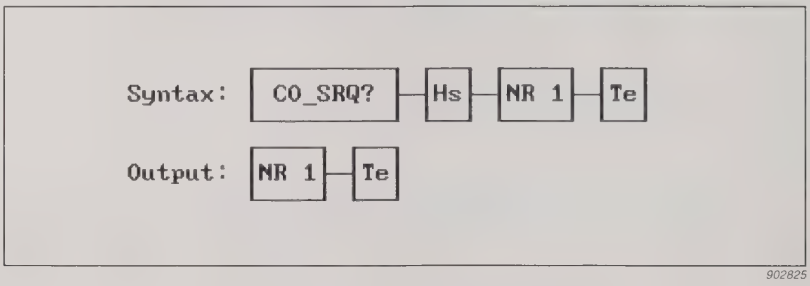


Fig. 4.12. Syntax for COntrol_SRQ?

Reads out the event number (range 0..255) for the event which has been registered in the Event Register specified in the message (either 1–4 or 8). The registers and the bit numbers corresponding to each event or state are shown in Tables 4.2 (a)–(e). For example, if you typed:

```
CO_SRQ? 8
```

this requests the number stored in register 8. If this is done, for example, after the SRQ button has been activated on the monitor, then the monitor should read out the decimal number 65 to the display as output. This is because bit numbers 1 and 7 are set as a result of the events "key pressed" and "SRQ button activated" ($2^0 + 2^6 = 65$)

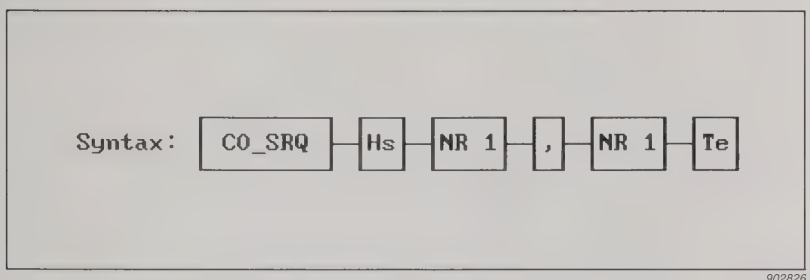


Fig. 4.13. Syntax for COntrol_SRQ

This message sets the Event Enable Register specified.
The first number (either 1–4 or 8) is a register number. The second number (range 0..255) is a bit-mask that determines which event can generate SRQ. For example, if you typed:

```
CO_SRQ 8,12
```

then the decimal number 12 is written into Register 8 (the Special Event Enable Register), setting bits 3 and 4, so that whenever the air pump is started or stopped a Service Request would be generated.

These registers are known to COntrol_SRQ:

Register 1 (Data Ready Register)	
Bit No	**Bit Name**
1	Sample Data Ready
2	Measurement Task Data Ready
3	Calibration Data Ready
4..8	*Not used*

T02609GB0

Table 4.2(a). The contents of Register 1

Register 2 (Reset Activated Register)	
Bit No	**Bit Name**
1..8	*Not used*

T02610GB0

Table 4.2(b). The contents of Register 2

Register 3 (Command Completed Register)	
Bit No	**Bit Name**
1	Input/Output Command Finished
2	Measurement Task Finished
3	Calibration Finished
4	SetUp Operation Finished
5	Memory Operation Finished
6	Print/Plot Finished
7..8	*Not used*

T02611GB0

Table 4.2(c). The contents of Register 3

Register 4 (Error Register)	
Bit No	**Bit Name**
1	Error In selftest detected
2..6	*Not used*
7	HW Error detected
8	SW Error detected

T02612GB0

Table 4.2(d). The contents of Register 4

Register 5
Register 6
Register 7
} Not used.

Register 8 (Special Event Register)	
Bit No	**Bit Name**
1	Key Pressed
2	Synchronize Required
3	Pump Started
4	Pump Stopped
5	Sample Measurement in progress
6	Bus Control Requested
7	SRQ Button Activated
8	Selftest Running

T02615GB0

Table 4.2(e). The contents of Register 8

4.2.10. **C**URRENT_**F**ILTER

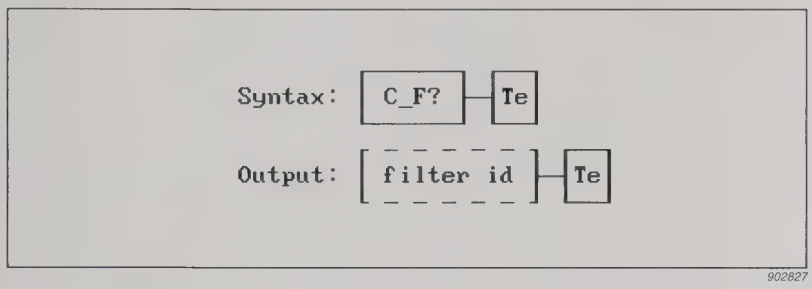


Fig. 4.14. Syntax for Current_Filter?

Reads out the "filter id". This identifies which filter (gas) is currently in use. For example, the monitor would read out "F_W" if the water vapour filter was the filter in use.

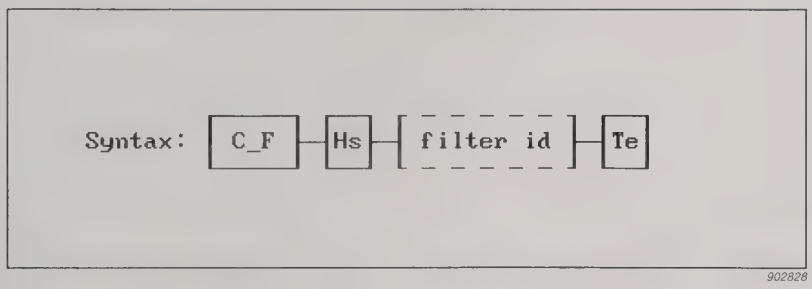


Fig. 4.15. Syntax for Current_Filter

Determines which filter (gas) will be displayed.

4.2.11. **CU**RRENT_**SE**TUP

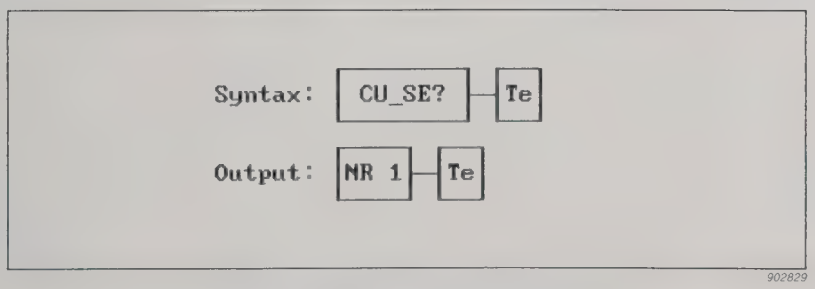


Fig. 4.16. Syntax for CUrrent_SEtup?

Reads out the number of the active "Monitoring Task" on the display. The output will be a number between 1 and 10 as it is possible to set up ten different monitoring tasks, but only one task may be active at any given time.

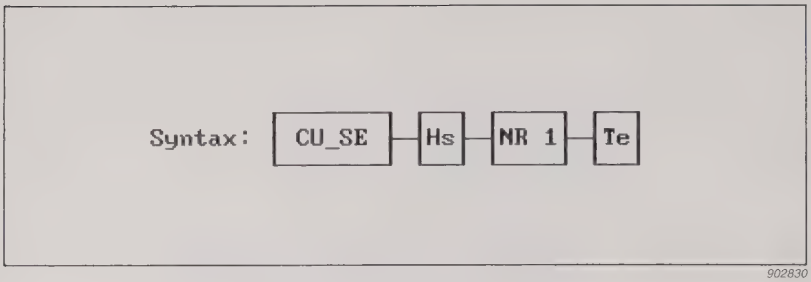


Fig. 4.17. Syntax for CUrrent_SEtup

Determines which "Monitoring Task" will be active.

4.2.12. **D**EFINE_**T**ERMINATOR

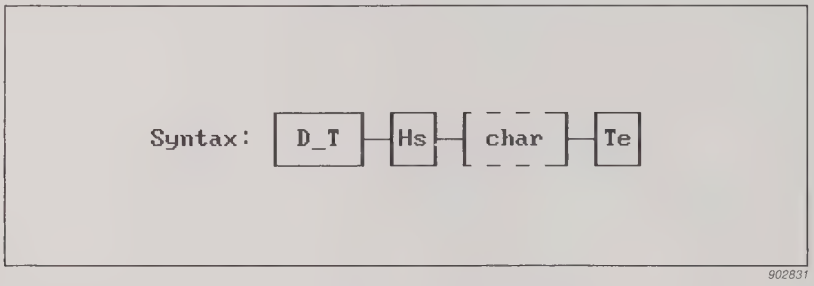


Fig. 4.18. Syntax for Define_Terminator

Defines the character that is used as the message terminator (Te). This message can only be used on the IEEE–488 interface. The message terminator for the RS–232–C interface cannot be changed and is fixed as the default character; line feed <LF>. This terminator may be redefined to any ASCII character decimal value between 1 and 31, with the exception of 13. For example, the given message might be "D_T 1", which defines the terminator as <SOH> because the <SOH> ASCII character has a decimal value of 1.

4.2.13. **D**ELETE_**M**EASUREMENT

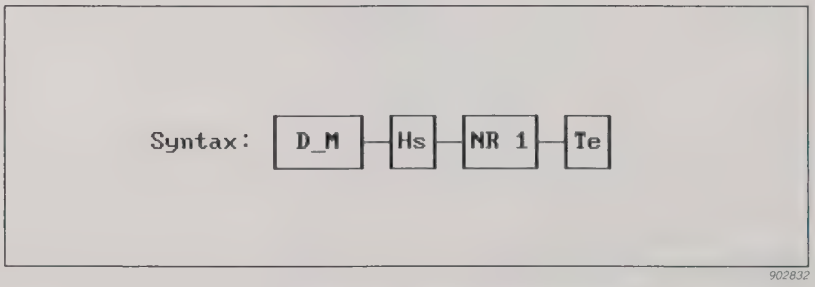


Fig. 4.19. Syntax for Delete_Measurement

Erases the contents of specified memory location (1..10) from the background memory.

4.2.14. **D**ISPLAY_**S**TATE

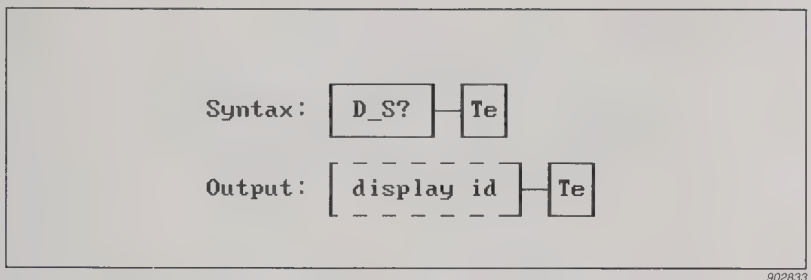


Fig. 4.20. Syntax for Display_State?

Indicates which display buffer is being displayed on the monitor. The output takes the form of a "display id". The display id options are:

POpUp	Used for error/status/reset messages.
REMote	The content is written via interface.
SYstem	Default display.

For example, the monitor will read out "SY" to the controller during normal system display.

/***** PROTECTED 1 *****/

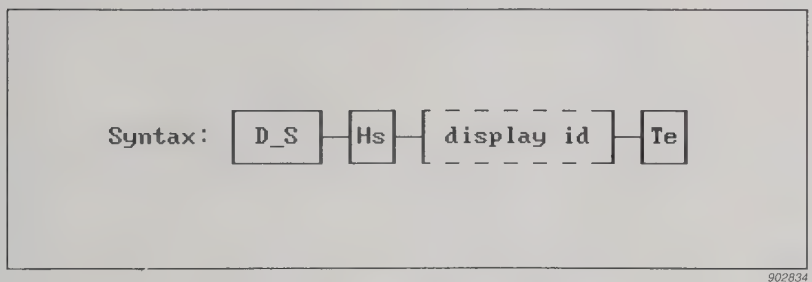


Fig. 4.21. Syntax for Display_State

Selects the display buffer to be read out on the monitor display.

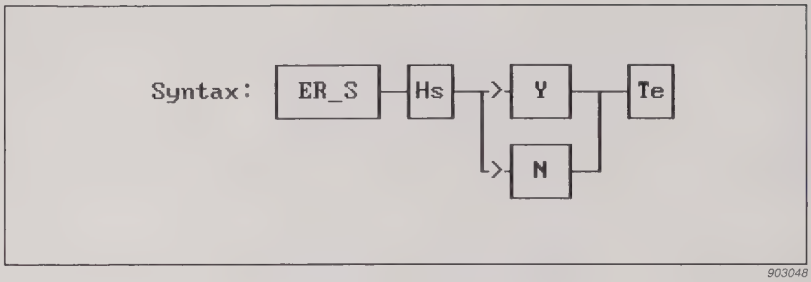


Fig. 4.22. Syntax for ERror_Stop

The **ER**ror_**S**top message is used to determine what information is displayed on the monitor screen when there is an error in the interface message sent by the user. The data field following the **ER_S** header can be **Y**es or **N**o, determining whether or not the **ER**ror_**S**top function is active.

Yes	Indicates error location .
No	Does not indicate error location.

For example, if the **ER**ror_**S**top function is set to **Y**es, and you were to type:

```
*IDN? 9
```

then the monitor screen would read:

```
TOO MUCH DATA
*IDN?
```

This is because no data is expected after the header, with this message, and the message is returned up to and including the character which forms the error, or start of the error.

Note: Only the minimum code is returned even when you send messages in full.

If the **ER**ror_**S**top function is set to **N**o, only the error message TOO MUCH DATA would be read out on the monitor screen.

Note: Any error in the interface message sent to the monitor will cause it to ignore all subsequent data. You must send an IFC **(Interface Clear)** bus command to reset, and the interface message must be sent again. IFC is an interface bus command and is defined in the interface standards.

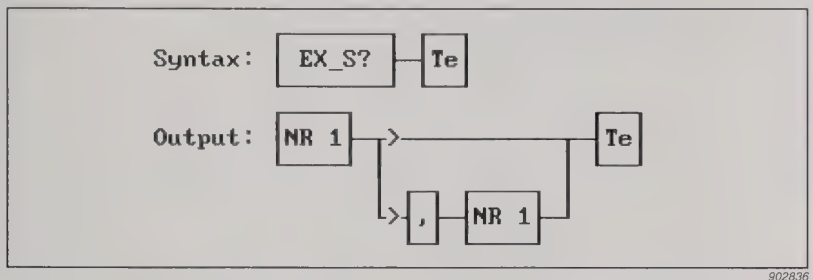


Fig. 4.23. Syntax for EXecute_State?

First integer gives a general description of the current status of the monitor.
Second integer is read out only if the first integer indicates that the monitor is being calibrated. It describes the calibration procedure. The meaning of the first integer read out is given in Table 4.3.

Output Number (NR 1)	Name	Description
0	Display_Run	Normal idle state
1	Edit_Run	Editing measurement
2	Function_Select	
3	Key_Seq_Entry	Entering key sequence
4	Key_Seq_Show	Displaying key sequence
5	Meas_Clean_up	Just after stop measurement
6	Meas_Count_Down	Waiting for time to run out
7	Meas_Init_Mode	Measurement mode selected
8	Meas_Run	Measurement running
9	Calib_Run	Calibration running
10	Memory_Run	Memory mode related
11	Plot_Print_Clean_Up	Just after end of plot/print
12	Plot_Print_Menu	Plot/print selected but not started
13	Plot_Run	Plot in progress
14	Print_Run	Print in progress
15	Setup_Run	Setup mode selected
16	Test_Mode	Test mode

T02752GB0

Table 4.3. The meaning of output from EXecute_State?

The second integer, given when the monitor is calibrating, indicates one of the following:

0 No error occurred.
1 Warning(s) detected.
2 Errors detected.

4.2.17. **E**XTENDED_**C**ONTROL

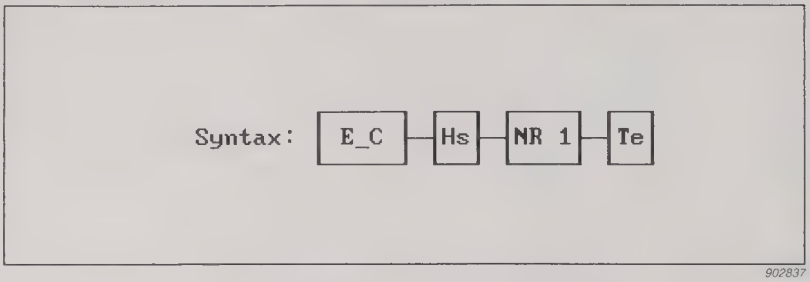


Fig. 4.24. Syntax for Extended_Control

This message is used to gain access to the protected messages. For example, if you were to type:

```
E_C 59
```

This would allow you to use the message **E**xternal_**P**ump to configure the monitor for an external pump (see section 4.2.18). Without using the **E**xtended_**C**ontrol message first, the monitor would not recognise this command.

Note:

You must send the **E**xtended_**C**ontrol message, before sending protected messages, every time a protected message is sent.

4.2.18. **E**XTERNAL_**P**UMP

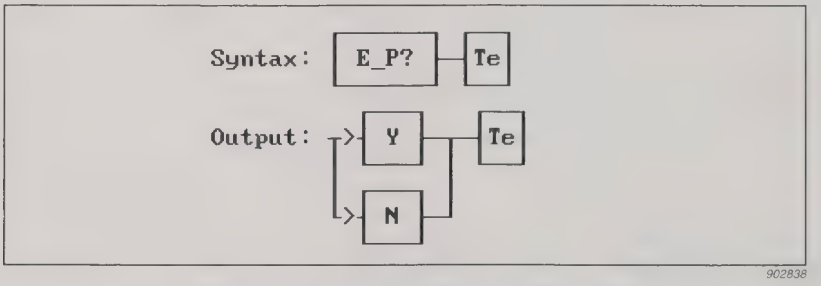


Fig. 4.25. Syntax for External_Pump?

Indicates whether the monitor is configured to work with an external pump, e.g. the pump in the Type 1303.

/***** PROTECTED 59 *****/

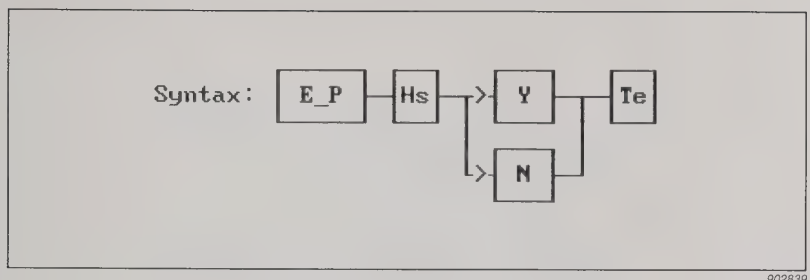


Fig. 4.26. Syntax for External_Pump

Configures the monitor to take an external pump into consideration when the sampling tube/analysis cell is to be rinsed. If an external pump is connected, the tube is rinsed independently of the monitor.

Yes	External pump connected
No	External pump not connected

When a measurement ends/stops, the monitor will set **E**xternal**_P**ump to No.

4.2.19. **IDEN**TIFICATION**?**

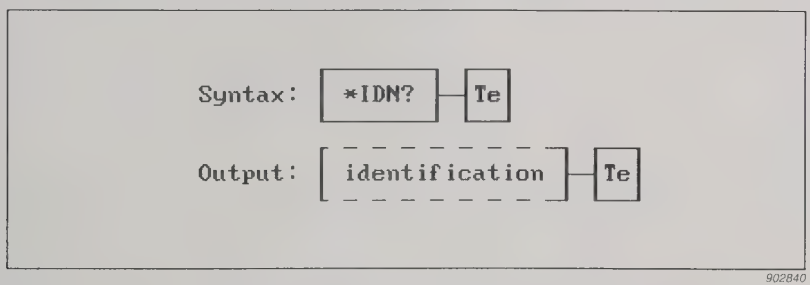


Fig. 4.27. Syntax for IDeNtification?

Gives instrument identification in the following format:

B&K,1302,ssss,VPxxxx.

ssss	string of max 10 characters containing the serial number.
xxxx	software number.

4.2.20. **ID**ENTIFY**?**

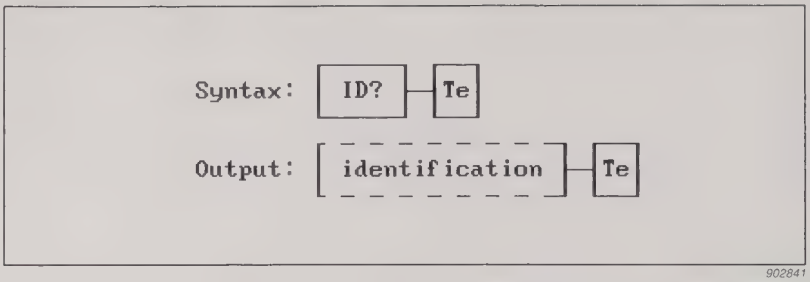


Fig. 4.28. Syntax for IDentify?

Gives instrument identification in this format:

B&K 1302

4.2.21. **K**EYBOARD_**L**OCKOUT

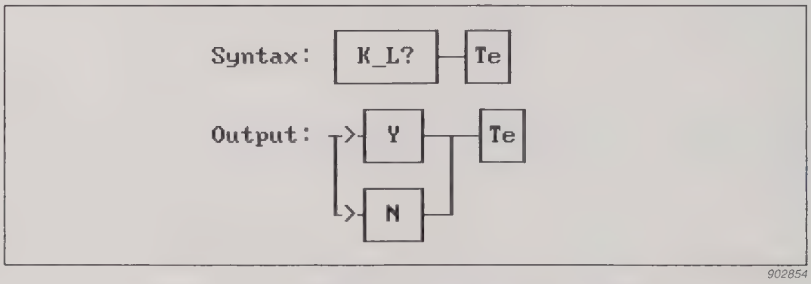


Fig. 4.29. Syntax for Keyboard_Lockout?

Indicates whether the front panel keys of the monitor are locked.

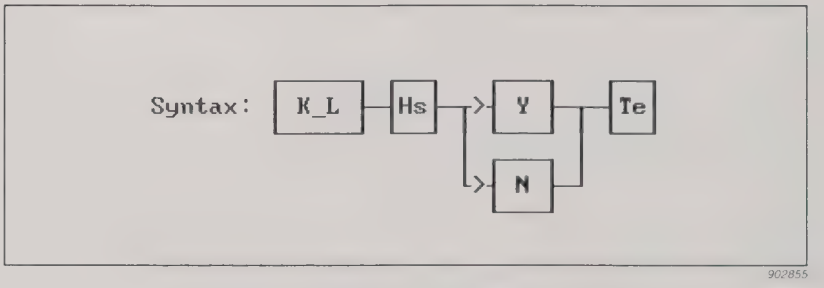


Fig. 4.30. Syntax for Keyboard_Lockout

Locks/unlocks the front panel push-buttons of the monitor.
The "**Stand by**" (or on/off) key still remains active, however, while the keyboard is "locked out".

4.2.22. **M**EASUREMENT_**S**TATE?

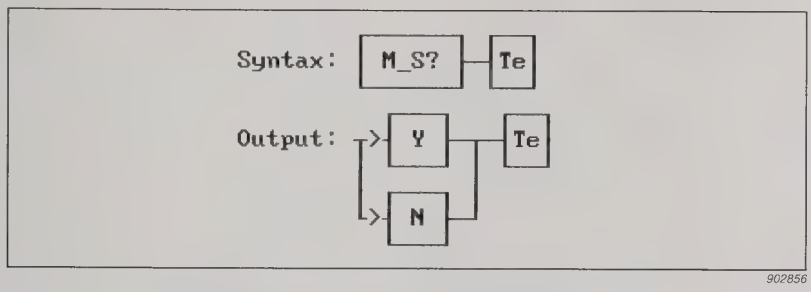


Fig. 4.31. Syntax for Measurement_State?

Indicates whether the monitor is measuring.

4.2.23. **O**UTPUT_**C**ALIBRATION_**D**ATA?

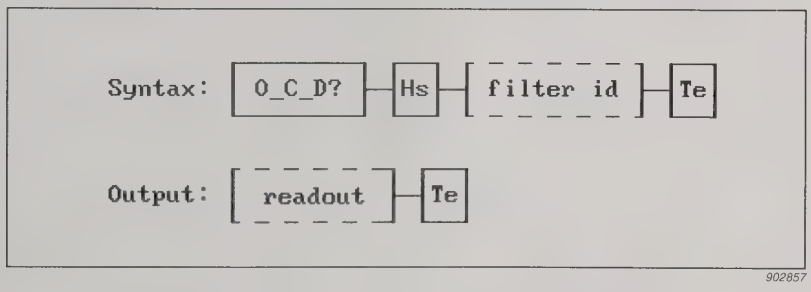


Fig. 4.32. Syntax for Output_Calibration_Data?

Reads out calibration data in ASCII form to the screen of the controller. "filter id" is one of the following:

ALl	Data for all filters
Filter_**A**	Data for filter A
Filter_**B**	Data for filter B
Filter_**C**	Data for filter C
Filter_**D**	Data for filter D
Filter_**E**	Data for filter E
Filter_**W**	Data for filter W

For further details, see Vol. 1 of the Type 1302 Instruction Manual.

4.2.24. **O**UTPUT_**E**RROR_**D**UMP?

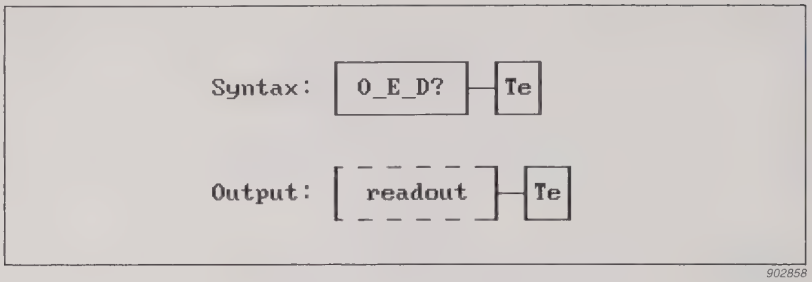


Fig. 4.33. Syntax for Output_Error_Dump?

Reads out memory data etc. in ASCII form. This message can be used to "dump" memory data to the controller screen after a fatal error has occurred in the monitor. The error can be generated from either hardware or software and causes an automatic restart (partial reset) of the monitor.

4.2.25. **O**UTPUT_**H**EADER

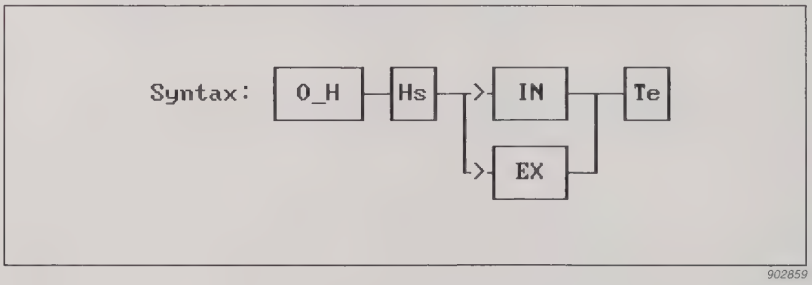


Fig. 4.34. Syntax for Output_Header

Allows you to define whether the monitor includes a message header with the readout messages. The default setting is without message header.

INclusive	Readout with message header.
EXclusive	Readout without message header.

For example, if you typed the message:

```
SE? 2, C_S
```

This asks the Type 1302 to indicate whether monitoring task 2 is set up for continuous sampling. If it is, there are two possible outputs depending on whether **O**utput_**H**eader is inclusive or exclusive. The output would be either (a) or (b) as given below:

(a) With **O**utput_**H**eader inclusive:

```
SE 2, C_S, Y
```

This is reusable as input.

(b) With **O**utput_**H**eader exclusive:

```
Y
```

4.2.26. **O**UTPUT_**M**EASUREMENT**?**

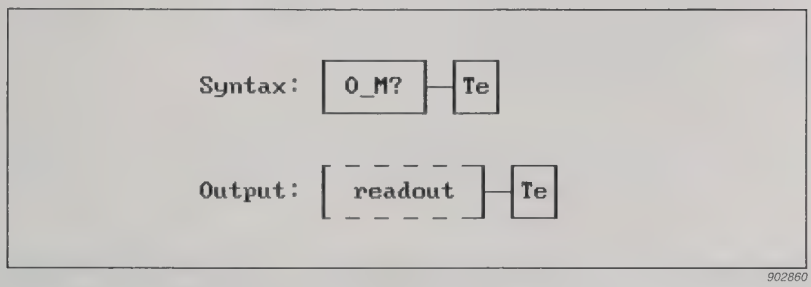


Fig. 4.35. Syntax for Output_Measurement?

Reads out measurement data in ASCII form to the screen of the controller.
For further details, see Vol. 1 of the Type 1302 Instruction Manual.

4.2.27. **O**UTPUT_**SE**TUP_**D**ATA**?**

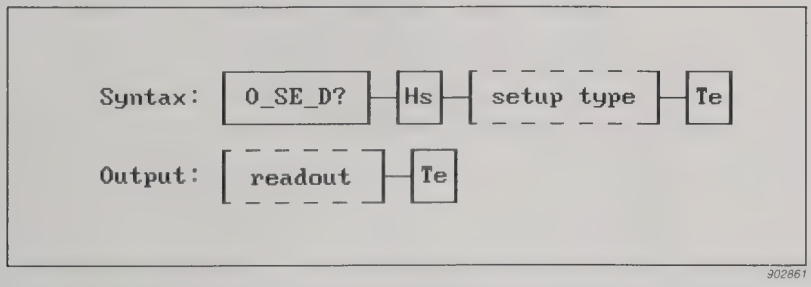


Fig. 4.36. Syntax for Output_SEtup_Data?

Reads out setup data in ASCII form to the screen of the controller. Setup Type is one of the following:

ALl
COMmunication
CONfiguration
ENvironment
FIlters
FOrmat
General
MEAsurement
SYstem
Units

These names refer to the headings in the "setup-tree". All data under these headings will be printed.

For further details, see Vol. 1 of the Type 1302 Instruction Manual.

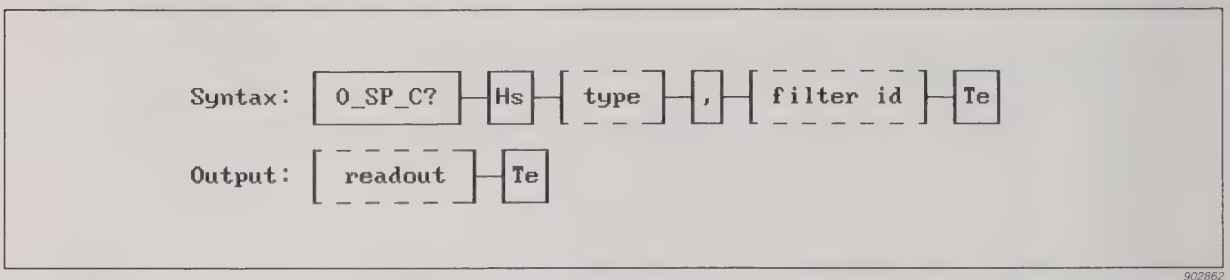


Fig. 4.37. Syntax for Output_SPecial_Compilation?

Reads out specially compiled data in ASCII form to the screen of the controller.
"Type" is **AP**plication. This is the Type 7620 Application Software used for remote control of the monitor.

"Filter id" is one of the following:

ALl	Data for all filters
Filter_**A**	Data for filter A
Filter_**B**	Data for filter B
Filter_**C**	Data for filter C
Filter_**D**	Data for filter D
Filter_**E**	Data for filter E
Filter_**W**	Data for filter W

The output format for each sample consists of the following:

[10 characters per gas] [3 error and warning bits] [Time stamp]

The 10 characters for each gas consist of the concentrations of each in the internal units, which are mg/m^3. This may or may not be the same units as the current units manually selected in the set up tree of the monitor. Dot marks (. . .) mean that the gas was not measured. A straight line (__) means that the monitor was unable to calculate the concentration of the gas. Of the error/warning bits, the first bit corresponds to gas feed errors/warnings, the second bit corresponds to any errors and the third bit corresponds to any warnings. The time stamp gives hours, minutes, seconds, in addition to the date.

4.2.29. **P**ASS_**C**ONTROL_**B**ACK

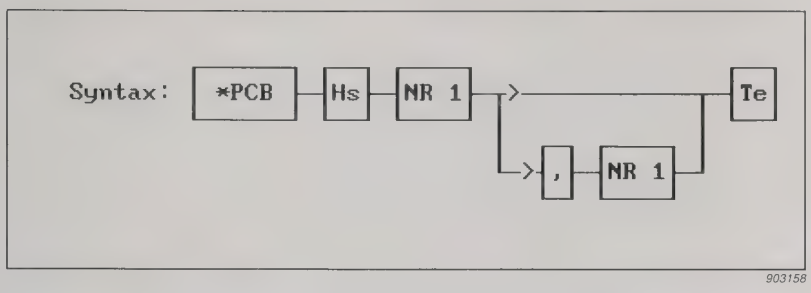


Fig. 4.38. Syntax for Pass_Control_Back

Determines the address to which control is to be passed back. NR_1 (0..30) is the bus address of the next controller. If both numbers are used, the first will be the primary and the second the secondary address of the next controller.

4.2.30. **PL**OT_**M**EASUREMENT

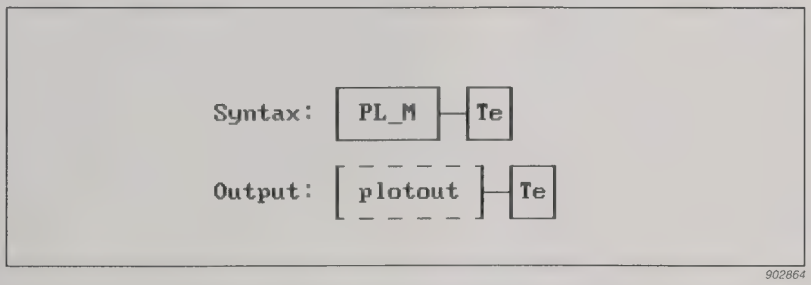


Fig. 4.39. Syntax for PLot_Measurement

The monitor sends a hard copy of the measurements to a plotter.
For further details, see Vol.1 of the Type 1302 Instruction Manual.

4.2.31. **P**OP_**U**P_**D**ISPLAY_**B**UFFER?

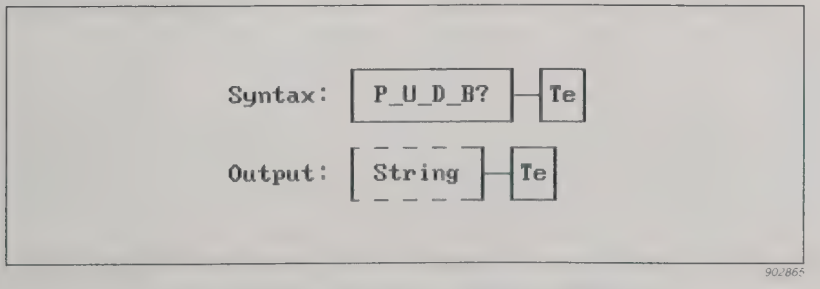


Fig. 4.40. Syntax for Pop_Up_Display_Buffer?

Reads out the contents of the Pop-up display. For example, the output string might be:

"UNKNOWN HEADER OR CHARACTER DATA"

4.2.32. **P**RINT_**C**ALIBRATION_**D**ATA

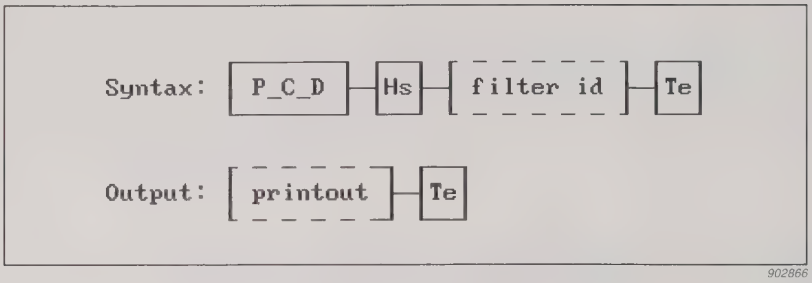


Fig. 4.41. Syntax for Print_Calibration_Data

Sends calibration data to a printer.
"filter id" is one of the following:

ALl	Data for all filters
Filter_**A**	Data for filter A
Filter_**B**	Data for filter B
Filter_**C**	Data for filter C
Filter_**D**	Data for filter D
Filter_**E**	Data for filter E
Filter_**W**	Data for filter W

For further details, see Vol. 1 of the Type 1302 Instruction Manual.

4.2.33. **P**RINT_**E**RROR_**D**UMP

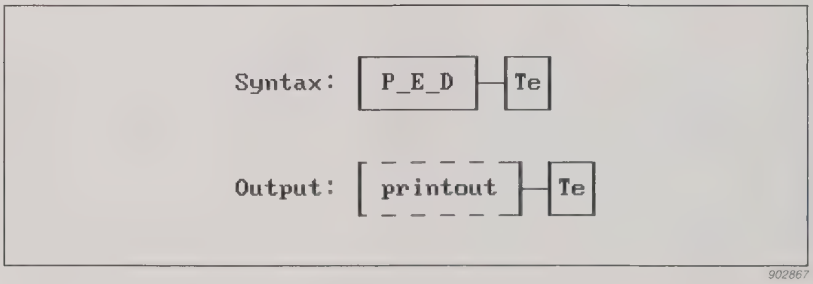


Fig. 4.42. Syntax for Print_Error_Dump

Sends memory data etc. to a printer in ASCII form. This message can be used to "dump" memory data after a fatal error has occurred in the monitor. The error can be generated from either hardware or software and causes an automatic restart (partial reset) of the monitor.

4.2.34. **PR**INT_**M**EASUREMENT

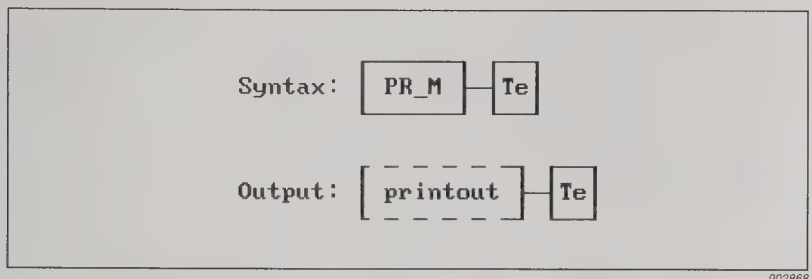


Fig. 4.43. Syntax for PRint_Measurement

Sends monitoring task data to a printer.
For further details, see Vol. 1 of the Type 1302 Instruction Manual.

4.2.35. **P**RINT_**SE**TUP_**D**ATA

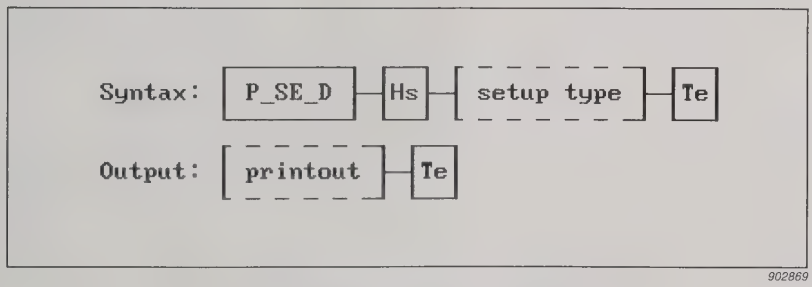


Fig. 4.44. Syntax for Print_SEtup_Data

Sends setup data to a printer.
Setup Type is one of the following:

ALl
COMmunication
CONfiguration
ENvironment
Fllters
FOrmat
General
MEAsurement
SYstem
Units

These names refer to the headings in the setup "tree". All data below these headings will be read out.

For further details, see Vol. 1 of the Type 1302 Instruction Manual.

4.2.36. **P**RINT_**SP**ECIAL_**C**OMPILATION

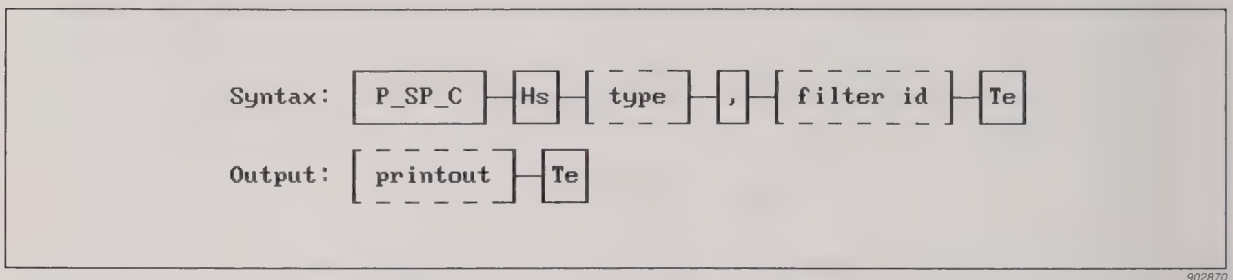


Fig. 4.45. Syntax for Print_SPecial_Compilation

Sends a printout of specially compiled data to a printer.
"Type" is **AP**plication. This is the Type 7620 Application Software used for remote control of the monitor.

For further information on filter id's and output format, see section 4.2.28 **O**UTPUT_**S**PECIAL_**C**OMPILATION?

4.2.37. **P**USH_**K**EY

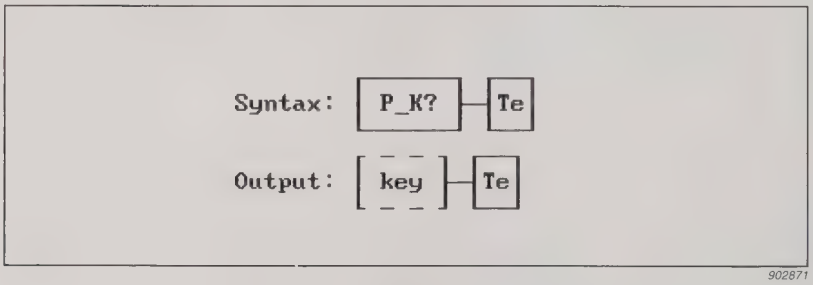


Fig. 4.46. Syntax for Push_Key?

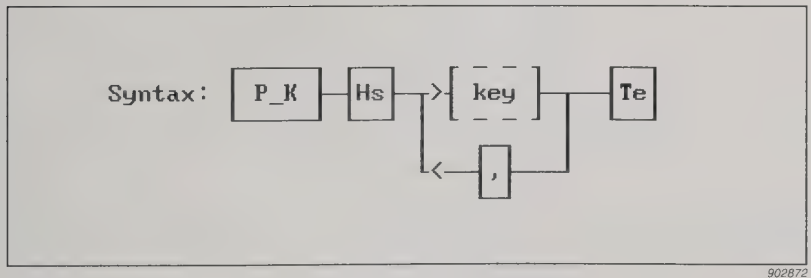
Indicates which front panel key has been the last activated. Key is one of the following:

Code	Key
AVerage	Average
Display_**N**ext	Next Display
Display_**P**revious	Previous Display
Clear_**E**ntry	Clear Entry
Default_**V**alue	Default Value
DOt	.
Enter_**E**xecute	Enter/Execute
Sign	—
Event_**M**ark	Event Marker
FUNction	Function
Go_**T**o_**H**ead	Go to Head
LOCal	Local
MEAsurement	Measurement
MEMory	Memory
Next_**G**as	Next Gas
PLot	Plot
PRint	Print
RESet	Reset
Select_**1**	S1
Select_**2**	S2
Select_**3**	S3
SEtup	Setup
SRQ	SRQ
STATUs	Status

T02614GB0

Table 4.3. The codes for Key

For example, the output would be "MEA" if the last key activated was the measurement key.



902872

Fig. 4.47. Syntax for Push_Key

Input of one or more front panel keys.

4.2.38. **R**ECALL_**M**EASUREMENT

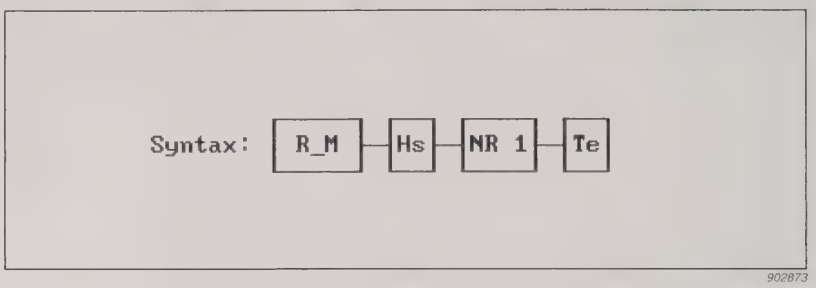


Fig. 4.48. Syntax for Recall_Measurement

Transfers measurement from the memory location specified (range 1..10) in the background memory into the display memory.

4.2.39. **R**ELAY

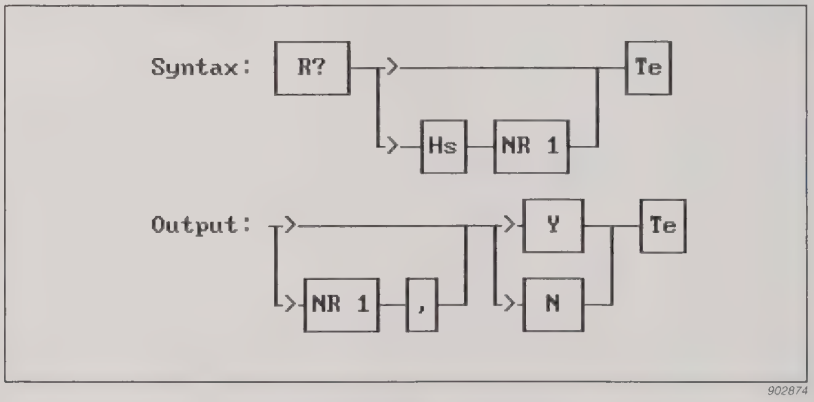


Fig. 4.49. Syntax for Relay?

Indicates the condition of/control over relay 1 and relay 2.

If the relay number is included, the condition of that relay will be read out by the monitor as below:

Yes	active
No	passive

If the relay number is left out, the readout indicates if the relays are being controlled via the interface.

/***** PROTECTED 59 *****/

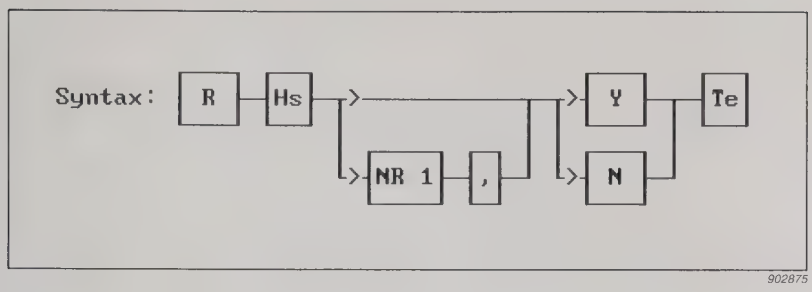


Fig. 4.50. Syntax for Relay

If the relay number is included, the condition of that relay will be set by the monitor as below:

Yes	active
No	passive

If the number is left out:

Yes	the relays will be controlled via the interface
No	the relays will be controlled by the monitor as normal alarm relays.

4.2.40. **R**EMOTE_**D**ISPLAY_**B**UFFER

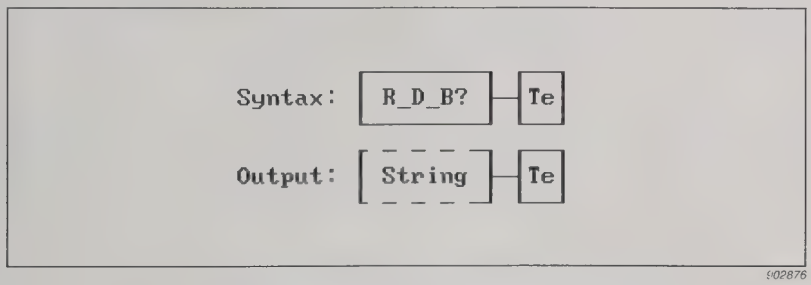


Fig. 4.51. Syntax for Remote_Display_Buffer?

Reads out the contents of Remote display. The output string will be whatever has been written into the buffer via the interface by the user.

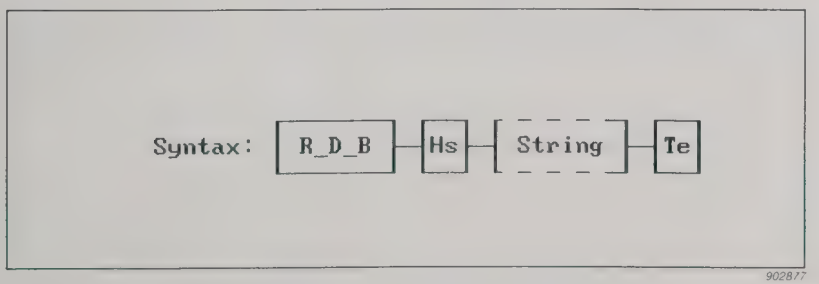


Fig. 4.52. Syntax for Remote_Display_Buffer

Reads data string into Remote display buffer, overwriting previous contents. The data string is 80 characters long and must be enclosed by double quotes (" ").

4.2.41. **RESET**_COMMAND

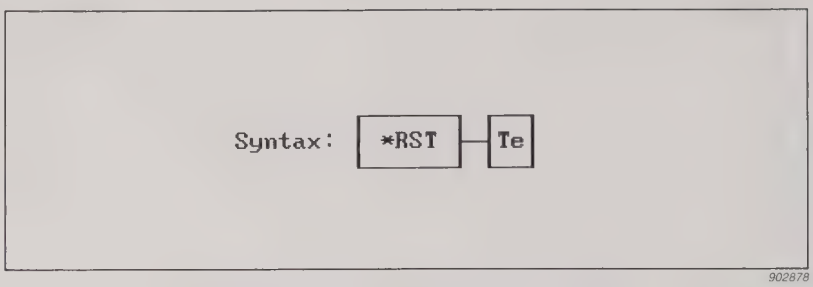


Fig. 4.53. Syntax for ReSeT_command

Partially resets the monitor. For further details, see Vol. 1 of the Type 1302 Instruction Manual, section 8.1.

4.2.42. **R**ESET_**S**TATUS_**B**YTE

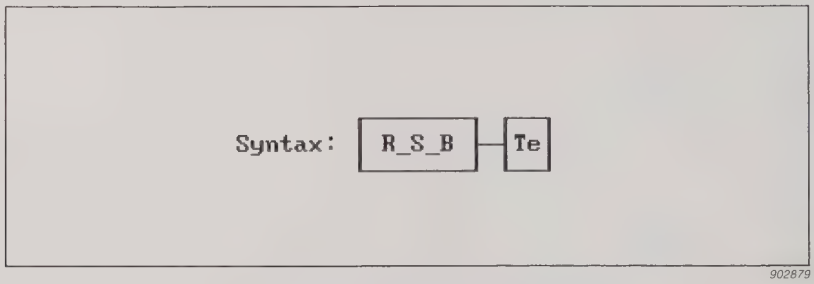


Fig. 4.54. Syntax for Reset_Status_Byte

Clears the Status byte. i.e. it is set to 00000000.

4.2.43. **RESET_SY**STEM

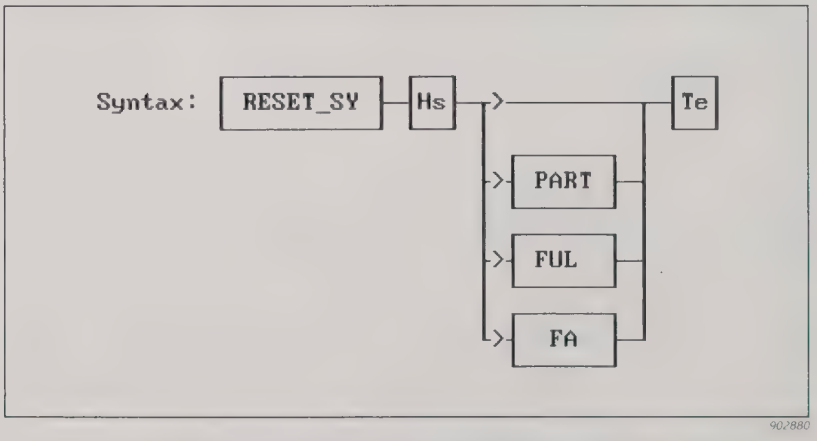


Fig. 4.55. Syntax for RESET_SYstem

Resets the system. The parameter should be one of these options:

PARTial	The monitor is partially reset.
FULl	The monitor is fully reset.

If parameter is omitted, the monitor is partially reset.

/***** PROTECTED 71 *****/

FActory	The monitor makes a total E2PROM reset.

For further details on resetting the monitor, see Vol. 1 of the Type 1302 Instruction Manual.

4.2.44. SELF_**TEST?**

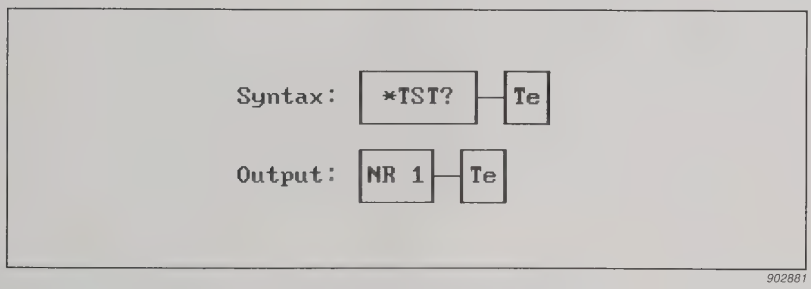


Fig. 4.56. Syntax for Self_TeST?

The monitor carries out a self-test. The number read out gives the result of this test.
The number has one of these values:

-2	The monitor is not in idle mode (display mode). Therefore the test cannot be performed.
-1	Operational Error Flags have been set after the test.
0	No flag has been set after the test.
1	Warning Flags have been set after the test.

For further details, see Vol. 1 of the Type 1302 Instruction Manual. Chapter 8 gives information on self-test and Appendix 19 gives information on Warning/Operational Error messages.

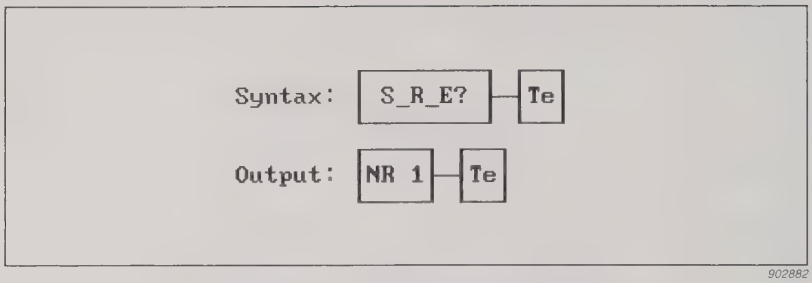


Fig. 4.57. Syntax for Service_Request_Enable?

The Service Request Enable Register is a mask for the Status Byte (see section 4.2.50) i.e. a bit which is set in the Status Byte will only cause a Service Request if the corresponding bit in the Service Request Enable Register is also set. This register is, therefore, used by the controller to enable and disable Service Requests.

This message will read out the decimal value of the SRQ enable register. The number is in the range 0..255. For example, if the register contained the binary number 00001000 and you typed:

```
S_R_E?
```

The monitor would read out the decimal number 8, indicating that bit 4 is set and that an error has occurred.

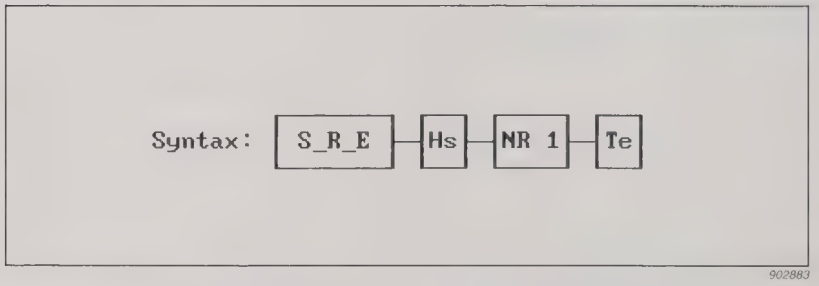


Fig. 4.58. Syntax for Service_Request_Enable

Sets the SRQ enable register. The number in the register lies in the range 0..255. For example, if you typed:

```
S_R_E 8
```

The monitor would set bit 4 of the register to binary 1 so that whenever bit 4 of the Status Byte is set (the Error bit) an SRQ is generated.

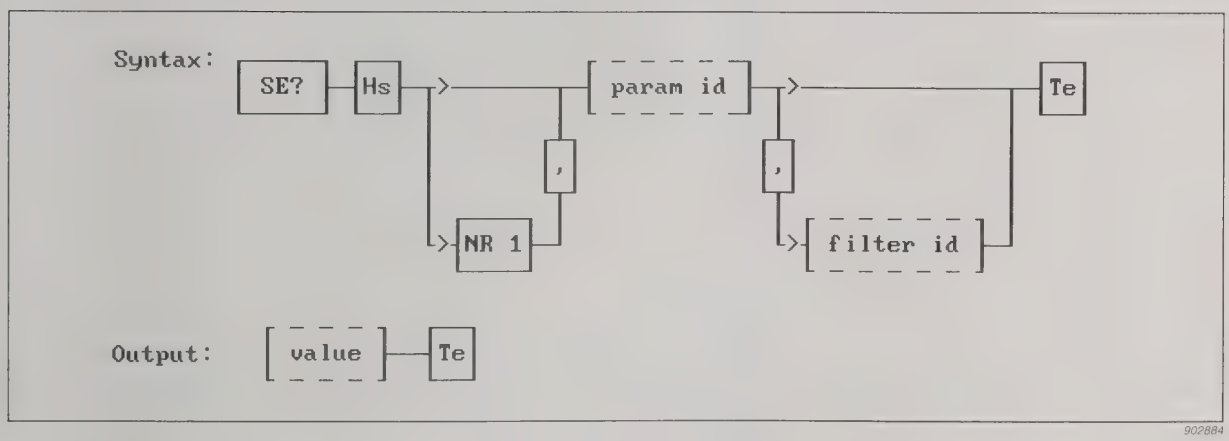
The contents of the Service Request Enable Register are given in the Table 4.4. See section 5.4.2 for further details.

Service Request Enable Register		
Bit No.	**Bit Value**	**Bit Name**
1	1	Data Ready
2	2	Reset Activated
3	4	Command Completed
4	8	Error
5	(16)	(Not User Definable)
6	32	Abnormal
7	(64)	(Not User Definable)
8	128	Special Event

T02753GB0

Table 4.4. The contents of the Service Request Enable Register.

4.2.46. **SE**TUP



902884

Fig. 4.59. Syntax for SEtup?

Reads out a Value parameter from the "set-up" of the monitor.

NR_1 (1..10)	is Setup Number
Param id	is the parameter identification.
Filter id	is the filter identification. e.g. F_A , F_W
Value	is numeric data (types NR1..3), character data, or string data.

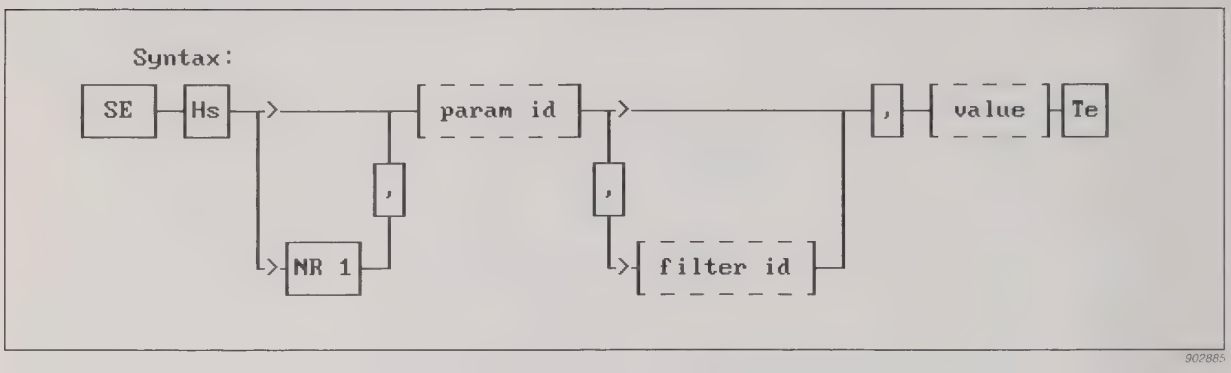


Fig. 4.60. Syntax for SEtup

Defines the specified part of the setup of the monitor.

NR_1 (1..10)	is Setup Number
Param id	is the parameter identification.
Filter id	is the filter identification. e.g. F_A , F_W
Value	is numeric data (types NR1..3), character data, or string data.

The following 4 examples show the use of the **SE**tup message with the parameter Value being of a different type in each case:

```
SE T_L, 10.5
```

In this case Value is numeric and the length of the sampling tube is set to 10.5 in the current length units.

```
SE S_H, X_O
```

In this case Value is character data .

```
SE G_N, F_A, "Carbon Dioxide"
```

In this case Value is a text string and the gas name for filter A is set to Carbon Dioxide.

```
SE 2 , M_P, "2 12:00"
```

In this case Value is a time string. Monitoring task No.2 is set up for a period of 2 days, and 12 hours.

All allowable Parameter id's and Values for **SE**tup are given in Tables 4.6(a) and (b).

Setup Parameter id's	Minimum Code
ALARM_LIMIT	A_L
AIR_PRESSURE	A_P
AVERAGE_TIME	A_T
BANK	B
BAUD_RATE	B_R
CLOCK	CL
CONCENTRATION_AXIS	C_A
CROSS_COMPENSATION	C_C
COLOR_MODE	C_M
CONTINUOUS_SAMPLING	C_S
CONCENTRATION_UNIT	C_U
DATE	DA
DATA_BIT	D_B
DATA_LOG	D_L
ERROR_LOGGING	E_L
GAS_NAME	G_N
GAS_WEIGHT	G_W
HARD_WIRE	H_W
HUMIDITY_UNIT	H_U
IEEE_1302_ADDRESS	I_1_A
IEEE_1302_CONTROLLER	I_1_C
IEEE_PLOTTER_ADDRESS	I_PL_A
IEEE_PRINTER_ADDRESS	I_PR_A
LENGHT_UNIT	L_U
MONITORING_MODE	M_M
MONITORING_PERIOD	M_P
NORMALIZATION_TEMPERATURE	N_T
PARITY	PARI
PLOTTER_LINK	PL_L
PEN_SPEED	P_S
PRESSURE_UNIT	PR_U
PRINTER_LINK	PR_L
PRINT_PLOT_ALL	P_P_A
PRINT_PLOT_FILTER	P_P_F
POWER_UP_TEST	P_U_T
STOP_BIT	S_B
SAMPLE_INTERVAL	S_I
SOFT HANDSHAKE	S_H
SOUND_VOLUME	S_V
TEMPERATURE_UNIT	T_U
TUBE_LENGTH	T_L
UA_NUMBER	UA_N
USE_FILTER	US_F
WATER COMPENSATION	W_C

T02768GB0

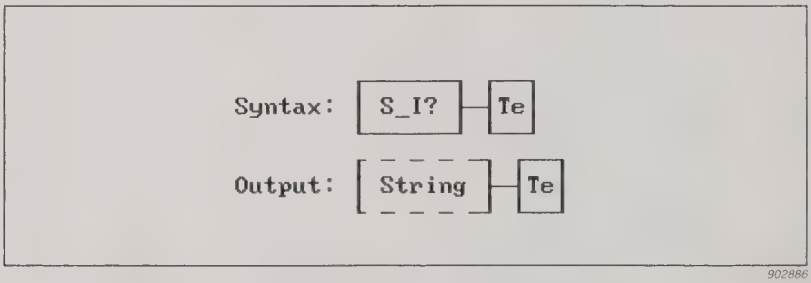
Table 4.6(a). Setup Parameter id's.

Setup Parameter "value"	Minimum Code
EVEN	EV
HARD_WIRE	H_W
IEEE 488	IE
LINEAR	LIN
LOGARITHMIC	LOG
LEASED_LINE	L_L
LEASED_COLOR	M_C
NO (NONE)	N
ODD	O
RS 232 C	RS
SINGLE_COLOR	SI_C
SWITCHED_LINE	S_L
THREE_WIRE	T_W
UNIT_C	U_CE
UNIT_F	U_FA
UNIT_ft	U_FEET
UNIT_K	U_KE
UNIT_kPa	U_kPa
UNIT_mbar	U_MBAR
UNIT_m	U_METER
UNIT_mg/m^3	U_MG_M
UNIT_mm Hg	U_MM HG
UNIT_ppm	U_PPM
UNIT_Tdew	U_TDEW
X-ON/X-OFF	X_O
YES	Y

T02767GB0

Table 4.6(b). Setup parameter Values.

4.2.47. **S**OFTWARE_**I**DENTIFICATION**?**



902886

Fig. 4.61. Syntax for Software_Identification?

Reads out the software identification string. For example, the output might be "1302 PROM ID VP217x 90-08-17 07.55".

4.2.48. **S**TART_**D**ELAYED_**M**EASUREMENT

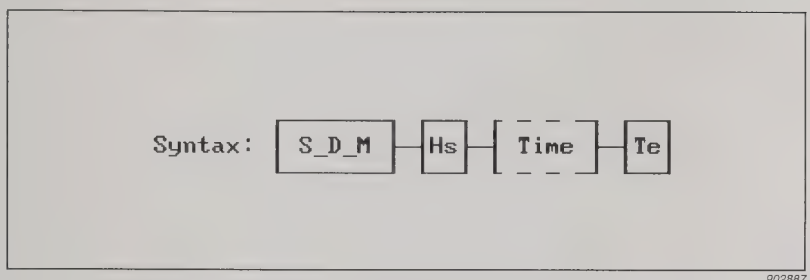


Fig. 4.62. Syntax for Start_Delayed_Measurement

Starts a monitoring task at a specified time.
"time" is in the range 00:00..23:59.

4.2.49. **STA**RT_**M**EASUREMENT

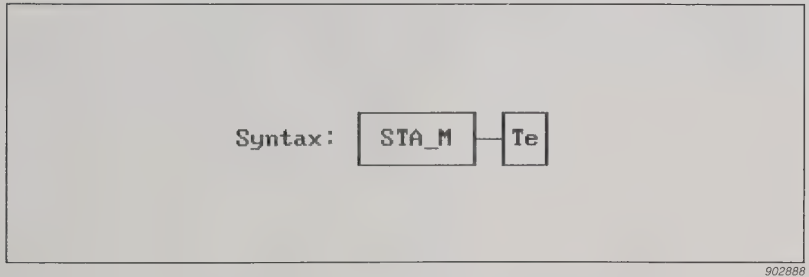


Fig. 4.63. Syntax for STArt_Measurement

The monitor starts a monitoring task immediately

4.2.50. **ST**ATUS_**B**YTE**?**

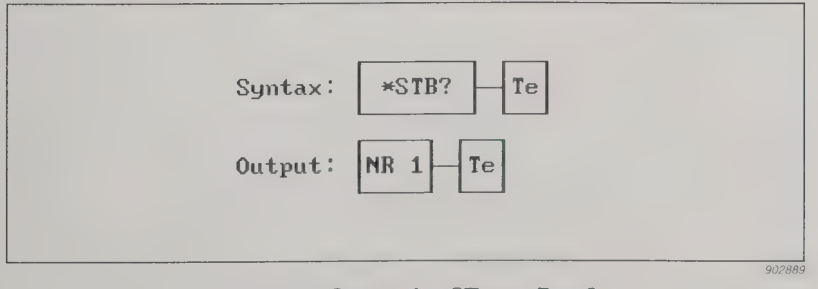


Fig. 4.64. Syntax for STatus_Byte?

Reads out the contents of the Status Byte. The number is in the range 0..255. If, for example, the Status Byte contained the binary number 00000001 and you typed:

```
*STB?
```

the monitor would return the decimal number 1, indicating that the Data Ready bit was set. See section 5.4.1 for further details.

Bit No.	Bit Name
1	Data Ready
2	Reset Activated
3	Command Completed
4	Error
5	BUSY
6	ABNORMAL
7	SRQ_line
8	Special Event

T02608GB0

Table 4.7. The contents of the Status Byte

4.2.51. **STOP_M**EASUREMENT

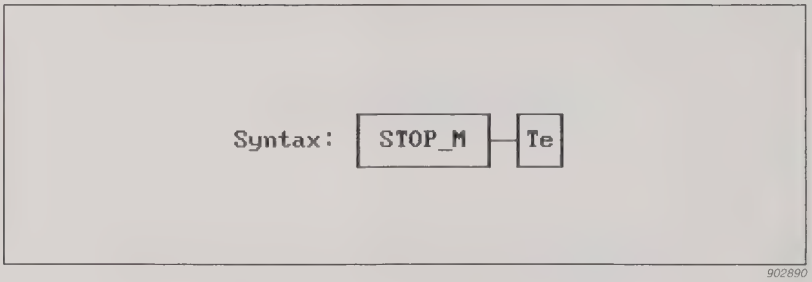


Fig. 4.65. Syntax for STOP_Measurement

Stops the current measurement cycle.

4.2.52. **STOR**E_**M**EASUREMENT

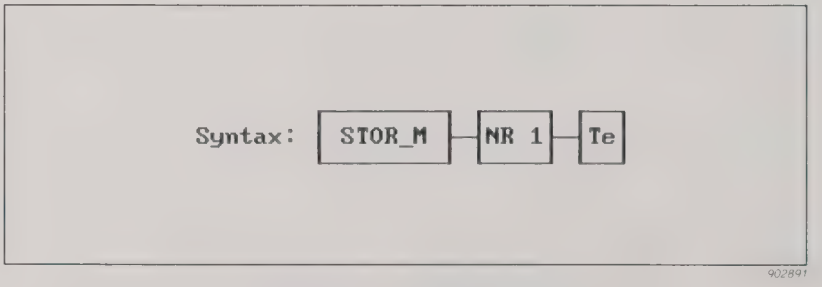


Fig. 4.66. Syntax for STORe_Measurement

Stores the contents of display memory in the background memory at the specified location (a number in the range 1..10).

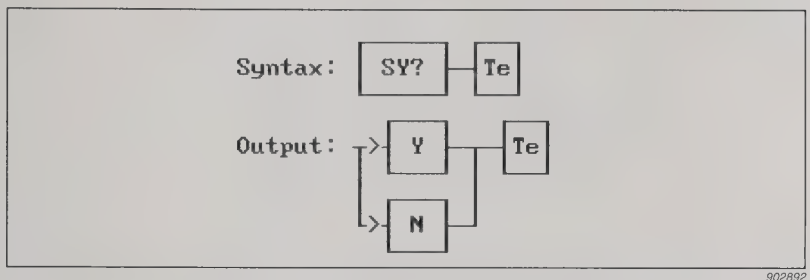


Fig. 4.67. Syntax for SYnchronize?

Reads out the pump synchronization condition. This message is for use when an external pump (e.g. a pump in the Type 1303) is connected to the monitor. The two possible parameters read out are:

Yes — Indicates that the monitor pump will start only when a synchronization command, **SY**NCHRONIZE, is received from the controller. This is the required condition when an external pump is connected to the monitor. If the synchronization condition is not set to **Y**es, and an external pump is connected, then airway errors can occur.

No — Indicates that the monitor pump is controlled by the monitor.

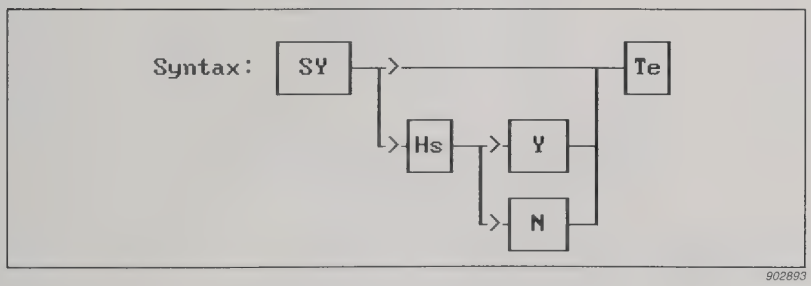


Fig. 4.68. Syntax for SYnchronize

/***** PROTECTED 59 *****/

Sets the pump synchronization condition to either **Y**es or **N**o:

Yes — the monitor pump will start only when a synchronization command, **SY**NCHRONIZE, is received from the controller.

No — the pump will be controlled by the monitor.

If the parameter (**Y**es or **N**o) is not used, the monitor pump starts if the monitor is waiting for the **SY**NCHRONIZE command.

4.2.54. **SY**STEM_**D**ISPLAY_**B**UFFER**?**

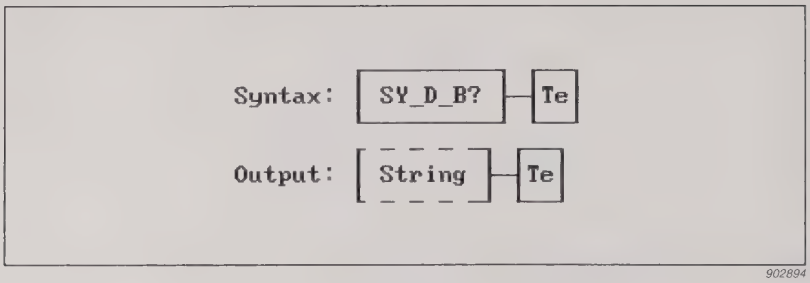


Fig. 4.69. Syntax for SYstem_Display_Buffer?

Reads out the contents of the monitor default display. For example, the output might be:

```
SELECT SET-UP BRANCH MEASUREMENT FORMAT CONFIGURATION
```

4.2.55. **T**IME_**S**INCE_**R**ESET**?**

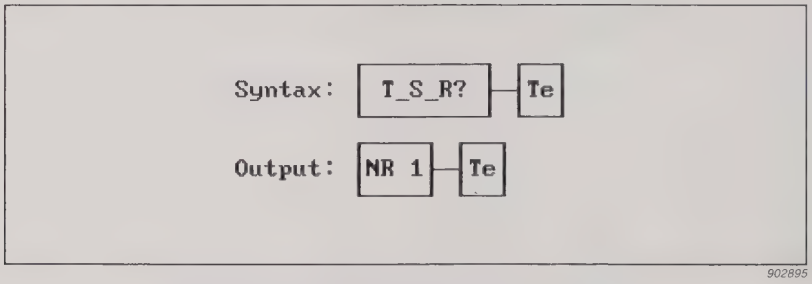


Fig. 4.70. Syntax for Time_Since_Reset?

Reads out the number of seconds since last reset/power-up of the monitor. The range is 0..4294967295.

4.3 THE IEEE–488 BUS MESSAGES

The IEEE–488 bus messages (IEEE_xxxx) can only be sent to the monitor via the RS–232–C interface. They enable the user to send commands to/receive information from the device(s) connected to the monitor via the IEEE–488 interface.

Notes

These messages can only be executed when the monitor is set-up as system controller. For further details, see Vol. 1 of the Type 1302 Instruction Manual, section 7.5.2.
As a system controller the controller is able to send device addresses, universal and addressed commands to other devices over the interface. To do this, however, it must send the ATN message over the interface (assert the ATN-line) before sending the actual message.

4.3.1. IEEE_ATTENTION

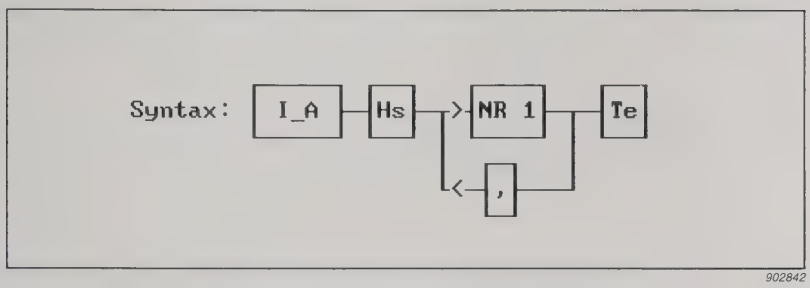


Fig. 4.71. Syntax for IEEE_Attention

Sends data on the IEEE-bus with the ATN-line asserted. The data consists of a maximum of 10 number (0..127) parameters, i.e. BYTE values.
This message enables control of devices on the IEEE-bus via RS–232–C.

EXAMPLE: I_A 63,95,49⟨Te⟩

This message will set the bus in the talker/listener idle state (UNL = 63 "111111", UNT = 95 "1011111") and addressed device 1 (49 "110001") as a listener.
The five least significant bits in each parameter correspond to the addresses in section 2.3, Table 2.1.

4.3.2. IEEE_CLEAR

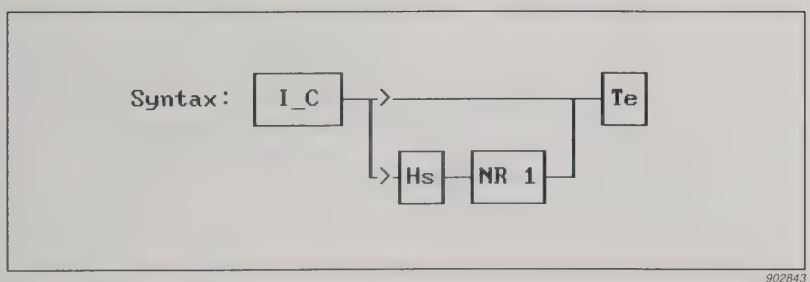


Fig. 4.72. Syntax for IEEE_Clear

If the number is left out the DCL (Device Clear) message is sent with ATN asserted.
If the number is included (0..30), a Listener address and SDC (Selected Device Clear) are sent.

4.3.3. IEEE_ENTER?

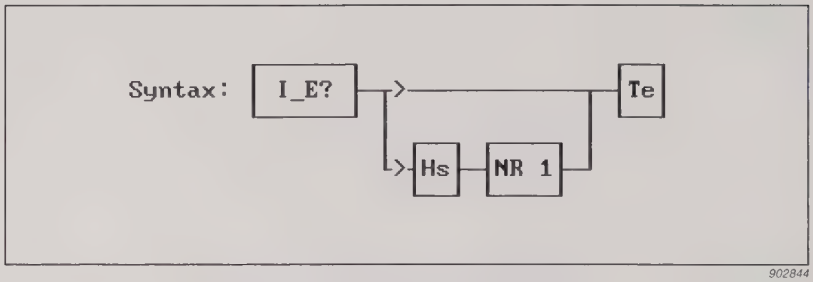


Fig. 4.73. Syntax for IEEE_Enter?

Reads in data from the IEEE bus and sends it on to the RS–232–C. If the number (0..30) is included, the IEEE bus is unaddressed and the specified device is addressed as a talker. UNL, UNT, Talk number-parameter are sent with ATN asserted, whereupon data is read in from this device.

4.3.4. IEEE_INTERFACE_CLEAR

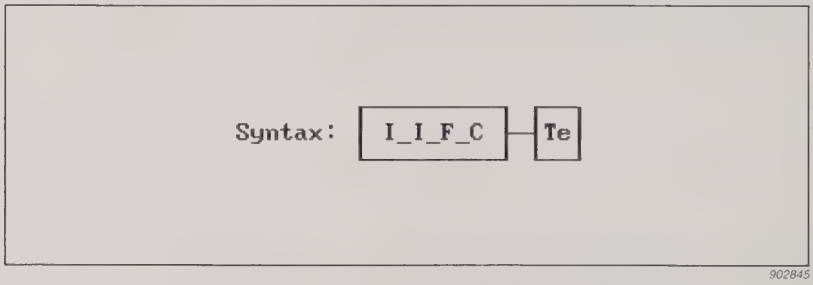


Fig. 4.74. Syntax for IEEE_InterFace_Clear

Sends IFC (Interface Clear) on the bus.

4.3.5. IEEE_LOCAL

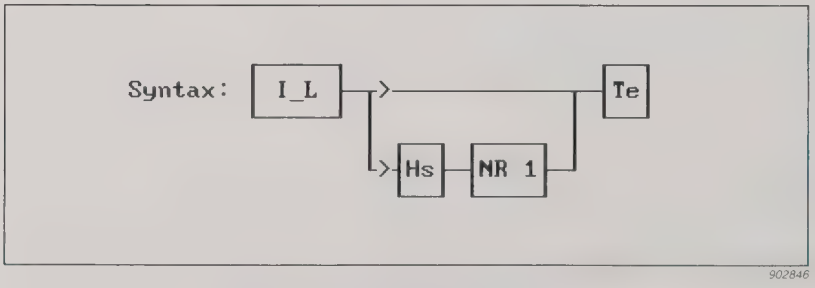


Fig. 4.75. Syntax for IEEE_Local

If the number (0..30) is included, UNL, the specified listener-address, and GTL (Go To Local) are sent with ATN asserted. The addressed device is set to Local State.

If the number is left out the REN (Remote Enable) line is unasserted (False). All devices on the bus are set to Local State.

4.3.6. IEEE_LOCAL_LOCKOUT

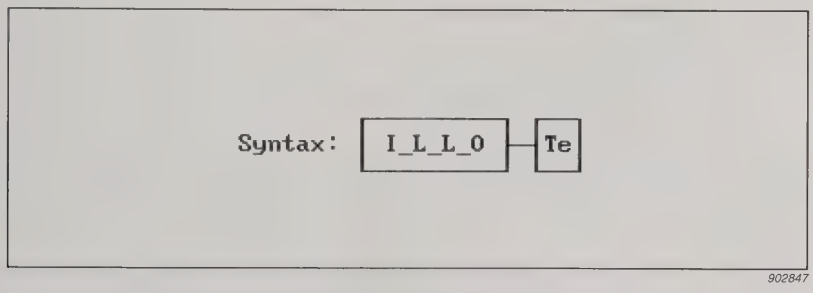


Fig. 4.76. Syntax for IEEE_Local_LockOut

Sends the LLO (Local Lock Out) message on the IEEE bus.
This command disables the return to local control (often a **"Local"** key) and thus prevents operation of the device via its front panel. The command is accepted regardless of whether the device is in local or remote mode.

4.3.7. IEEE_OUTPUT

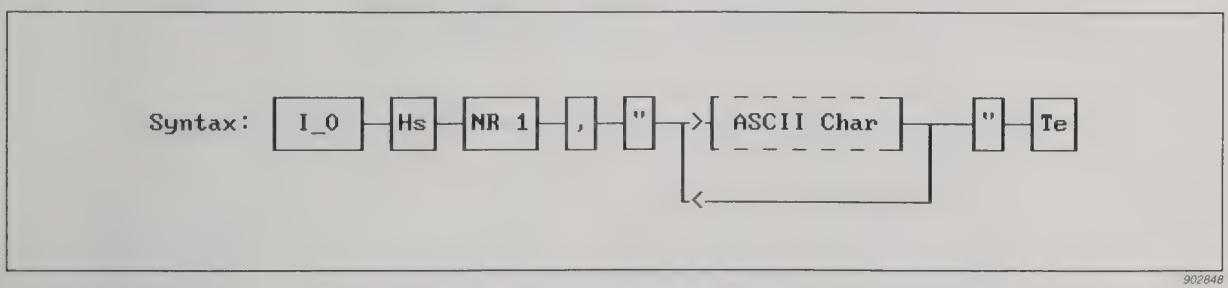


Fig. 4.77. Syntax for IEEE_Output

Sends data on the IEEE bus. If the number (0..30) is included, the specified device is addressed as a listener. The "ASCII char" is sent as normal data on the bus.

4.3.8. **I**EEE_**P**ARALLEL_**P**OLL**?**

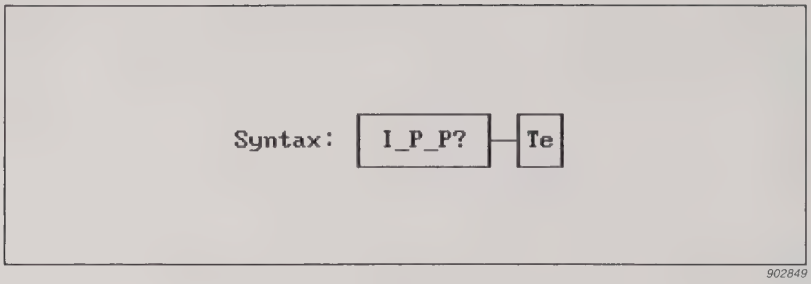


Fig. 4.78. Syntax for IEEE_Parallel_Poll?

Initiates a Parallel Poll sequence on the IEEE bus. The response is returned via the RS–232–C (Byte value).

4.3.9. **I**EEE_**PPC** (IEEE_Parallel_Poll_Configure)

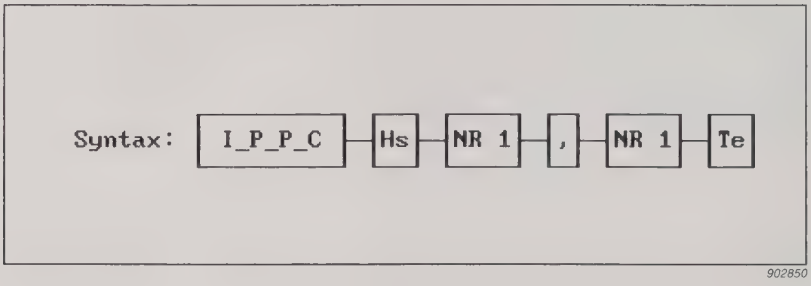


Fig. 4.79. Syntax for IEEE_Parallel_Poll_Configure

Executes a configuration of the Parallel Poll answer from a specified device.

First number parameter gives the address (0..30).

Second number parameter gives the configuration value (0..15). The three least significant bits indicate which bit position for the specified device the reply should be directed to. The fourth, least significant, bit indicates whether the reply should be "1" or "0" for a logical TRUE PP-status.

4.3.10. IEEE_**PPU** (IEEE_Parallel_Poll_Unconfigure)

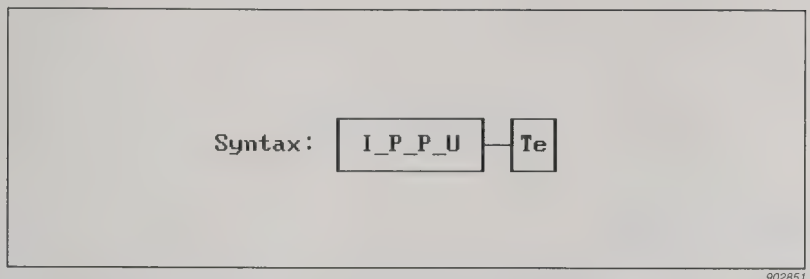


Fig. 4.80. Syntax for IEEE_Parallel_Poll_Unconfigure

Sends out PPU (Parallel Poll Unconfigure) with ATN asserted.

4.3.11. IEEE_**R**EMOTE

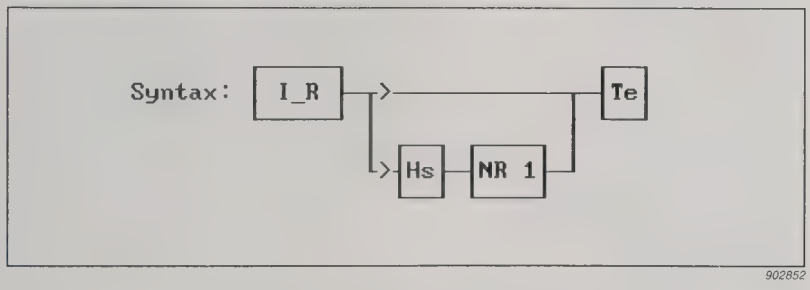


Fig. 4.81. Syntax for IEEE_Remote

Asserts the REN (Remote Enable) line. This places the addressed device in the Remote State, i.e. it can be controlled only via the IEEE–488 bus and not via the front panel.

4.3.12. IEEE_**S**ERIAL_**P**OLL?

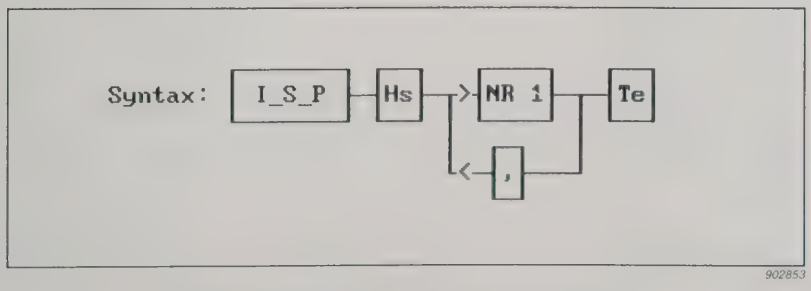


Fig. 4.82. Syntax for IEEE_Serial_Poll?

Performs a Serial Poll on up to 5 connected devices. The result for each device is returned as an 8–bit binary word.

5. APPENDIX

5.1. INTERFACE CHARACTER DATA

Alphabetical list of interface characters data	
Data Fields	**Minimum code**
ALL	AL
APPLICATION	AP
AVERAGE	AV
ALARM_LIMIT	A_L
AIR_PRESSURE	A_P
AVERAGE_TIME	A_T
A_D_VALUES	A_D_V
BANK	B
BAUD_RATE	B_R
CLOCK	CL
COMMUNICATION	COM
CONFIGURATION	CON
CONCENTRATION_AXIS	C_A
CROSS_COMPENSATION	C_C
CALIBRATION_DATA	C_D
CLEAR_ENTRY	C_E
CURRENT_FILTER	C_F
COLOR_MODE	C_M
CONTINUOUS_SAMPLING	C_S
CONCENTRATION_UNIT	C_U
DATE	DA
DOT	DO
DATA_BIT	D_B
DATA_LOG	D_L
DEFAULT_VALUE	D_V
DISPLAY_NEXT	D_N
DISPLAY_PREVIOUS	D_P
ENVIRONMENT	EN
EVEN	EV
EXCLUSIVE	EX
ENTER_EXECUTE	E_E
EVENT_MARK	E_M
ERROR_LOGGING	E_L
ERROR_VALUES	E_V
FACTORY	FA
FORMAT	FO
FULL	FUL
FUNCTION	FUN
FILTERS	FI
FILTER_A	F_A
FILTER_B	F_B
FILTER_C	F_C
FILTER_D	F_D

102882GB0

Table 5.1(a). List of interface character data

Data Fields	Minimum code
FILTER_E	F_E
FILTER_W	F_W
GENERAL	G
GAS_NAME	G_N
GAS_VALUES	G_V
GAS_WEIGHT	G_W
GO_TO_HEAD	G_T_H
HARD_WIRE	H_W
HUMIDITY_UNIT	H_U
IEEE 488	IE
INCLUSIVE	IN
IEEE_1302_ADDRESS	I_1_A
IEEE_1302_CONTROLLER	I_1_C
IEEE_PLOTTER_ADDRESS	I_PL_A
IEEE_PRINTER_ADDRESS	I_PR_A
LINEAR	LIN
LOCAL	LOC
LOGARITHMIC	LOG
LEASED_LINE	L_L
LENGTH_UNIT	L_U
MEASUREMENT	MEA
MEMORY	MEM
MEASUREMENT_DONE	M_D
MONITORING_MODE	M_M
MONITORING_PERIOD	M_P
MULTI_COLOR	M_C
NO (NONE)	N
NEXT_GAS	N_G
NORMALIZATION_TEMPERA...	N_T
ODD	O
PARITY	PARI
PARTIEL	PART
PLOT	PL
PRINT	PR
PLOTTER_LINK	PL_L
PEN_SPEED	P_S
POP_UP	PO_U
PRESSURE_UNIT	PR_U
PRINTER_LINK	PR_L
PROCESS_DATA	PR_D
PUMP_DONE	PU_D
PUSH_KEY	P_K
PRINT_PLOT_ALL	P_P_A
PRINT_PLOT_FILTER	P_P_F
POWER_UP_TEST	P_U_T
REMOTE	REM
RESET	RES
REVERSE	REV
RS 323 C	RS
REP_NEXT	R_N
REP_PREVIOUS	R_P
REP_S1	R_S1
REP_S3	R_S3
SETUP	SE
SIGN	SI
SRQ	SR
STANDARD	STAN

T02883GB0

Table 5.1(b). List of interface character data, cont.

Data Fields	Minimum code
STATUS	STATU
STATISTICS	STATI
SYSTEM	SY
SELECT_1	S_1
SELECT_2	S_2
SELECT_3	S_3
STOP_BIT	S_B
SAMPLE_DATA	SA_DA
SAMPLE_DONE	SA_DO
SAMPLE_INTERVAL	S_I
SETUP_DATA	SE_D
SINGLE_COLOUR	SI_C
SLUT_CALIBRATION	SL_C
SLUT_MEASUREMENT	SL_M
SOFT_HANDSHAKE	S_H
SOUND_VOLUME	S_V
SWITCHED_LINE	S_L
TEMPERATURE_UNIT	T_U
THREE_WIRE	T_W
TUBE_LENGTH	T_L
UNITS	U
UA_NUMBER	UA_N
UNIT_C	U_CE
UNIT_F	U_FA
UNIT_ft	U_FEET
UNIT_K	U_KE
UNIT_kPa	U_KPA
UNIT_mBar	U_MBAR
UNIT_m	U_METER
UNIT_mg/m^3	U_MG_M
UNIT_mmHg	U_MMHG
UNIT_ppm	U_PPM
UNIT_Tdew	U_TDEW
USE_FILTER	US_F
WARNING_VALUES	W_V
WATER_COMPENSATION	W_C
X-ON/X-OFF	X_O
YES	Y

T02606GB0

Table 5.1(c). List of interface character data, cont.

5.2. PROGRAMMING EXAMPLE

The program given below is written in a Hewlett Packard version of BASIC; Version 4.0. This is because unlike other versions of BASIC, the language itself supports the IEEE—488 interface directly. Hewlett Packard BASIC can only be used on Hewlett Packard machines, typically in this case calculators such as the HP9820. It first instructs the Type 1302 to begin a monitoring task (which, it is assumed, has already been set up). It then waits until the sample data is ready and prints it out, repeating this procedure until the entire measurement task has been completed. For these purposes 2 loops are used, one within the other.

```
  5 REM              <* Line 10 defines the 2 string variables and  *>
  6 REM              <* sets their maximum size in bytes. Line 20   *>
  7 REM              <* defines the variable "Event" as an integer. *>

 10 DIM String$[90],Event$[80]
 20 INTEGER Event

 25 REM              <* The next line assigns a name to the         *>
 26 REM              <* interface address 714 (the device address   *>
 27 REM              <* of the Type 1302 is taken to be the default *>
 28 REM              <* value (14) in this case).                   *>

 30 Bk1302=714

 35 REM              <* The next line clears the Type 1302          *>
 36 REM              <* interface.                                  *>

 40 Clear Bk1302
 50 Print "Start"

 55 REM              <* The next line sends the interface message   *>
 56 REM              <* to the Type 1302 to start measuring         *>
 57 REM              <* according to the current set up, using the  *>
 58 REM              <* interface message "STArt Measurement".      *>

 60 OUTPUT Bk 1302:"STA_MEAS"

 65 REM              <* The first loop entered is for the whole     *>
 66 REM              <* measurement task.                           *>

 70 LOOP

 75 REM              <* The second loop entered is for each         *>
 76 REM              <* individual sample.                          *>

 80    LOOP

 90 REM              <* The next line sends the "COntrol Service    *>
 91 REM              <* ReQuest" interface message to the Type 1302 *>
 92 REM              <* which reads the contents of (in this case)  *>
 93 REM              <* Event Register No.1.                        *>

100       OUTPUT Bk1302:"CO_SRQ? 1"

105 REM              <* The next line reads the number output from  *>
106 REM              <* the Type 1302 into the string variable      *>
107 REM              <* "Event$".                                   *>

110       ENTER Bk1302:Event$

115 REM              <* The next line converts the string to an     *>
116 REM              <* integer.                                    *>

120       Event=VAL(Event$)
```

```
122 REM              <* The next line means wait 2 seconds.          *>

124       WAIT 2

127 REM              <* The next line tests to see whether the       *>
128 REM              <* Sample Data Ready bit (bit 0) is set (equal *>
129 REM              <* to 1). If it is set, then the system exits   *>
130 REM              <* the second loop entered. Note that in HP     *>
131 REM              <* BASIC 4.0 the bits are numbered 0 - 7        *>
132 REM              <* whereas in the rest of the manual they have *>
133 REM              <* been arbitrarily numbered 1 - 8.             *>

135       EXIT IF BIT(Event;0)=1
140     END LOOP

141 PRINT "Get Result"

145 REM              <* The next line sends the interface message    *>
146 REM              <* "Output SPecial Compilation" to              *>
147 REM              <* to the Type 1302 which then outputs the      *>
148 REM              <* sample data.                                 *>

150 OUTPUT Bk1302;"O_SP_C? AP;ALL

155 REM              <* The next line reads the data string output   *>
156 REM              <* from the Type 1302 into the  string          *>
157 REM              <* variable "String$".                          *>

160 ENTER Bk1302:String$

161 REM              <* The next line prints the sample data in the *>
162 REM              <* format shown in section 4.2.28.              *>

163 PRINT "Result is  ":String$

164 REM              <* The next line tests to see whether the       *>
165 REM              <* Measurement Task Data Ready bit              *>
166 REM              <* (bit 1) is set (equal to 1) . If it is,      *>
167 REM              <* then the measurement task has been           *>
168 REM              <* been completed and the system exits the      *>
169 REM              <* first loop entered.                          *>

          170 EXIT IF BIT(Event,1)=1
171 END LOOP

175 REM              <* The next line instructs the Type 1302 to     *>
176 REM              <* stop the current measurement                 *>
177 REM              <* cycle using the interface message            *>
178 REM              <* "STOP Measurement".                          *>

180 OUTPUT Bk1302:"STOP_M"
190 END
```

5.3. ERROR MESSAGES

Error Message	Description
1302 IS NOT CONTROLLER IN CHARGE	The monitor must have control of the bus, but the monitor is not the CIC (Controller In Charge) The monitor should be set up as "System Controller" or have control transferred to it.
1302 IS NOT SYSTEM CONTROLLER	The monitor must be the "System Controller" for the message to be performed. It is not sufficient if the monitor is the CIC.
BAD BLOCK SPECIFICATION SYNTAX	Syntax error. Illegal specification of the block data. The correct syntax is #nc..c, where n is the length count and c is the digit. The size of the block may be wrong.
BAD CHARACTER DATA SYNTAX	Syntax error. Illegal character found in the character data types.
BAD HEADER DATA SYNTAX	Syntax error. Illegal character found in the message header.
BAD NR1 SYNTAX	Syntax error. Illegal NR1 numeric definition. Refer to "IEEE–488.2 – 1987 Standard" section 8.7.2 for the correct NR1 syntax.
BAD NR2 SYNTAX	Syntax error. Illegal NR2 numeric definition. Refer to "IEEE–488.2 – 1987 Standard" section 8.7.3 for the correct NR2 syntax.
BAD NR3 SYNTAX	Syntax error. Illegal NR3 numeric definition. Refer to "IEEE–488.2 – 1987 Standard" section 8.7.4 for the correct NR3 syntax.
BAD STRING DATA SYNTAX	Syntax error. The text string is not surrounded by quotes (″).
BAD TERMINATOR	Syntax error. Illegal message terminator, i.e. not <LF> or the one specified with DEFINE_TERMINATOR.
BAD TIME STRING SYNTAX	Syntax error. The text string to indicate date/time is not correct.
BLOCK DATA EXPECTED	Syntax error. Illegal data type received.
BLOCK DATA NOT COMPLETED	Syntax error. Block data ended/interrupted before the usual termination.
BLOCK DATA OVERRUN	Syntax error. Block data has not been ended according to the block specification.
BLOCK DATA. PARAMETER OUT OF RANGE	The data in the block is illegal.
CHARACTER DATA EXPECTED	Syntax error. The wrong data type was received.
CONTROLLER JOB ABORTED	Time Out on a job where the monitor is the controller. The usual monitor controller functions, i.e. Plot or Print, will not cause this error. It may happen in connection with messages from the RS–232 that are sent on via the IEEE–488.
CONTROLLER NOT ACTIVE	The monitor is supposed to be the controller, in connection with a certain function like Print or Plot. But the System Controller (computer) has taken back the control which it had temporarily passed to the monitor.
HANDSHAKE ERROR	Occurs in connection with the RS–232 <–> IEEE–488 conversion.
IEEE ABORTED	In the RS–232 <–> IEEE–488 conversion communication, communication on the IEEE–488 side has stopped.
IEEE CONTROL RECEIVED/NOT REQUESTED	The monitor wishes to send print/plot/error-/data-log via the IEEE–488 without being the CIC (Controller In Charge). Before the monitor gains control, output is aborted by the user. The monitor now takes control, however it is not necessary now for the monitor to be the controller.

T02886GB0

Table 5.2(a). List of error messages

Error Message	Description
IEEE CONTROL TIMEOUT	The monitor wishes to send print/plot/error-/data-log via the IEEE–488 without being CIC (Controller In Charge). If there are no other controllers on the bus, the monitor should be the "System Controller". CIC can not hand over control to the monitor.
IEEE CONTROL TIMEOUT/NOT REQUESTED	The monitor wishes to send print/plot/error-/data-log via the IEEE–488 without being CIC (Controller In Charge). Before the monitor gains control, output is aborted by the user. If there are no other controllers on the bus, so the monitor should be the "System Controller". CIC can not hand over control to the monitor.
IEEE ERROR	Error in the data from the IEEE–488 in connection with the RS–232<–>IEEE–488 conversion.
IEEE-INTERFACE LOCKED FROM RS232	The internal communication channel of the monitor is being used by the RS–232 interface. Communication with the monitor via the IEEE–488 is locked out by RS–232.
ILLEGAL CHARACTER	Illegal syntax. The error can appear in many contexts.
ILLEGAL CHARACTER DATA	The character data is not correct in this context.
ILLEGAL STATE FOR OPERATION	The monitor is currently in an operating state where the message is illegal.
ILLEGAL STRING DATA	The contents of the string is not correct.
INTERNAL BUFFER OVERFLOW	Syntax error. The Header/Character data is too big. The text string is too long etc.
JOB IS CANCELLED	The current output-message is interrupted by a new input-message.
JOB NOT ALLOWED FOR INPUT OR OUTPUT	Syntax error. The user has changed an input-only message into an output message by use of "?", or an output-only message into an input message.
MISSING CHARACTER	E.g. a "+" and <LF> are received. The "+" indicates that a numerical value (NRx) is to follow, but this numerical value does not appear.
NO CONTROL PASS BACK ADDRESS	When the monitor has to perform the print/plot functions, and it is not the "System Controller", control has to be passed to it by the active controller (computer). Afterwards, the return address must be known (via the message *PCB). This error occurs if the return address (to pass back control) is unknown.
NOT ALLOWED	
NR1 DATA EXPECTED	Syntax error. The wrong data type received.
NR1 OR CHARACTER DATA EXPECTED	Syntax error. The wrong data type received.
NR2 DATA EXPECTED	Syntax error. The wrong data type received.
NR3 DATA EXPECTED	Syntax error. The wrong data type received. Refer to "IEEE–488.2 – 1987 Standard" section 8.7.4 for the correct NR3 syntax.
NR3 EXPONENT OVERFLOW	Syntax error. The wrong data type received. Refer to "IEEE–488.2 – 1987 Standard" section 8.7.4 for the correct NR3 syntax.
NR3 OVERFLOW +/- 1.0E12	The exponent was too big. The monitor internal limit was exceeded.
OUTPUT DATA CANCELLED	The current output-message is interrupted by a new input message.
PARAMETER EXPECTED	Syntax error. The parameter for the message is missing.
PARAMETER OUT OF RANGE	The numerical parameter is illegal.
RECEIVE ERROR	A fundamental error in receiving the message.

T02885GB0

Table 5.2(b). List of error messages, cont.

Error Message	Description
RELAY CONTROL NOT ENABLED	The user is trying to control the alarm relays of the monitor and access to the relays has not been opened.
RS232-INTERFACE LOCKED FROM IEEE	The internal communication channel of the monitor is being used by the IEEE–488 interface. Communication with the monitor via the RS–232 is locked out by IEEE–488*.
STRING DATA EXPECTED	Syntax error. Illegal data type received.
STRING LENGTH OUT OF RANGE	The string is longer than allowed.
TIME OUT ON PCB	Control has temporarily been handed over to the monitor, but the System Controller (computer) does not want it passed back.
TOO MUCH DATA	Syntax error. There are more parameters in the message than necessary. Otherwise, the message is correct and will usually be executed.
UNKNOWN HEADER OR CHARACTER DATA	Syntax error. Unknown message header and character data.
WARNING: CTS HANDSHAKE CONFLICT	The monitor receives a handshake signal on the RS–232 CTS line without being set up for using CTS. There is a handshake conflict between the monitor and the device connected to the RS–232.
WARNING: DSR HANDSHAKE CONFLICT	The monitor receives a handshake signal on the RS–232 DSR line without being set up for using DSR. There is a handshake conflict between the monitor and the device connected to the RS–232.
WARNING: MEMORY FULL WARNING: MEMORY LOCATION ALREADY USED WARNING: MEMORY LOCATION EMPTY	Errors in connection with the use of the Background Memory.
WARNING: NO DATA FOR PLOT OR PRINT	There is not sufficient data in the monitor to make an output (plot/print).
WARNING: NOT ENOUGH DATA FOR PLOT	There is not sufficient data in the monitor to make a plot-out.
WARNING: RS232 FRAMING ERROR. DATA LOST	The monitor was receiving corrupted data on the RS–232 RX line.
WARNING: RS232 OVERFLOW. DATA LOST	The monitor receives data on the RS–232 RX line faster than it can handle. The internal buffer is full.
WARNING: RS232 OVERRUN. DATA LOST	The monitor receives data via the RS–232 RX line faster than it can handle.
WARNING: RS232 PARITY ERROR. DATA LOST	The monitor receives data with illegal parity.
WARNING: SET-UP AND/OR CALIBRATION CONFL	The monitor is about to start a measurement, but the set-up of the monitor and the possibilities of the current calibration are in conflict.
WARNING: TIME OUT. DEVICE NOT CONNECTED	The monitor wishes to send print/plot/error-/data-log via IEEE–488, but there is no response from any device on the address being sent. The device may not be properly connected or it may have have been given an incorrect address in the monitor set-up.
WARNING: TIME OUT. DEVICE NOT READY	The monitor wishes to send print/plot/error-/data-log. The transmission starts, but the device stops reception of data for too long. The device may be off-line. The device may be too slow. The device may no longer be properly connected.
WARNING: X-ON/X-OFF HANDSHAKE CONFLICT	The monitor receives the X–On/X–Off signal via RS–232 without being set up to use X–On/X–Off. There is a handshake conflict between the monitor and the device connected to the RS–232.
WRONG DATA TYPE	

T02884GB0

Table 5.2(c). List of error messages, cont.

5.4. INTRODUCTION TO SERVICE REQUESTS

Under certain conditions it is necessary for the monitor to demand attention from the controller or otherwise interrupt the normal sequence of interface communication. The monitor does this by raising the "Service Request" (SRQ) line of the interface. Service Requests can therefore be used for device synchronisation. The SRQ function is commonly used to synchronise the operation of the monitor with an external computer which has IEEE–488 or RS–232–C controller capabilities. Typically, SRQ indicates that data is ready to be transmitted or an error condition exists. The monitor generates a Service Request which interrupts the computer. When interrupted, the computer performs a serial poll on the monitor Status Byte and resets the SRQ line. The computer is then free to perform other tasks while waiting for a signal from the monitor.

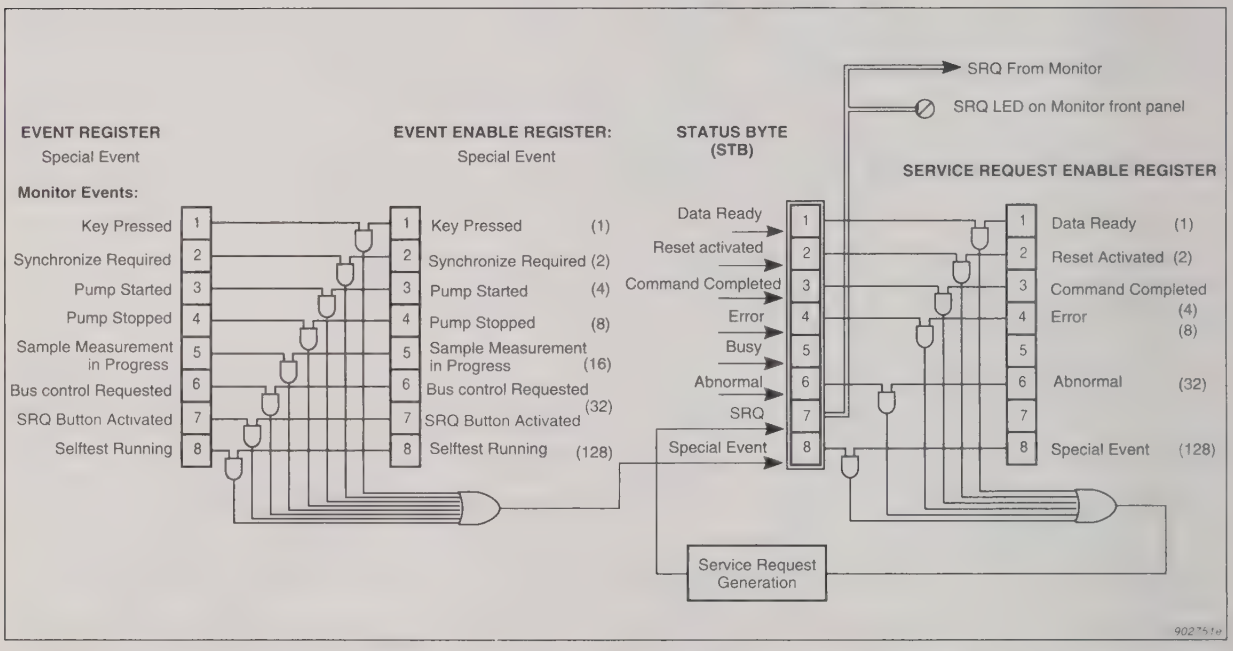


Fig. 5.1. Logic diagram of the "Special Event" Event register, Event Enable register, Status Byte and Service Request Enable register.

Fig. 5.1 shows part of the logic diagram for generating Service Requests. The input to 5 of the 8 bits of the Status Byte (bits 1–4 and 8) come from the 5 pairs of Event Registers and Event Enable Registers. One of these 5 pairs of Event Registers and Event Enable Registers (in this case the "Special Event" Registers) is shown.

- The Event Enable Registers act as masks for the Event Registers.
- The Service Request Enable Register acts as a mask for the Status Byte.

5.4.1. The Status Byte

The Status Byte has 8 bits which are by default set to 0. Seven of the bits are only set when specific conditions occur; the eighth bit (bit 7) is used to signal that the monitor has set the SRQ line. For example, when a command has been completed, bit 3 in the Status Byte is set (see Fig 5.1). The Status Byte can be read using the message **ST**ATUS_**B**YTE? (see section 4.2.50). It can also be read out as a part of a serial poll sequence. There is no interface message enabling you to change the contents of the Status Byte.

5.4.2. The Service Request Enable Register

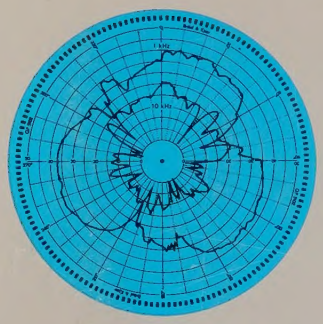
The Service Request Enable Register is a mask for the Status Byte. An event which causes a bit to be set in the Status Byte will not alone cause the SRQ line to be raised. If, however, the corresponding bit in the Service Request Enable Register has also been set, then the Status Byte bit will cause the SRQ line to be raised.

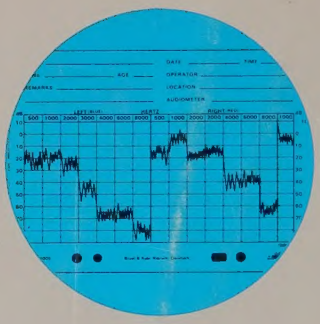
All bits in the Service Request Enable Register have a default value of 0. You can read out the contents of the register using the **S**ERVICE_**R**EQUEST_**E**NABLE? message, and you can define the bit pattern by using the **S**ERVICE_**R**EQUEST_**E**NABLE message. This determines which conditions cause the monitor raising the SRQ line. (see section 4.2.45).

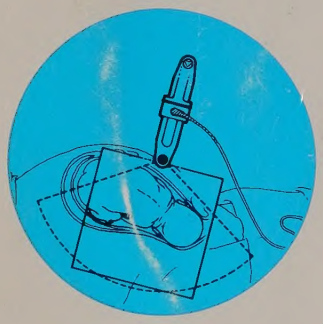
5.4.3. The Event and Event Enable Registers

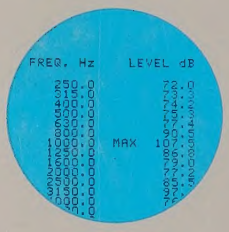
The input to 5 of the 8 bits of the Status Byte (bits 1-4 and 8) come from the 5 pairs of Event Registers and Event Enable Registers. These pairs of registers are linked together in a similar way to the Status Byte and Service Request Enable Register. 20 bits from the 5 Event Registers are used to indicate specific events. Each of these 20 bits has a corresponding bit in an Event Enable Register. As with the Status Byte, a corresponding Event Enable Register bit must be set before an event can generate an output. The contents of the 5 Event Registers are given in Tables 4.2(a)–(e), see section 4.2.9. Using the **CO**ntrol_**SRQ** message you can define the contents of the Event Enable Register specified. Using the **CO**ntrol_**SRQ**? message you can read the contents of the Event Enable Register specified.

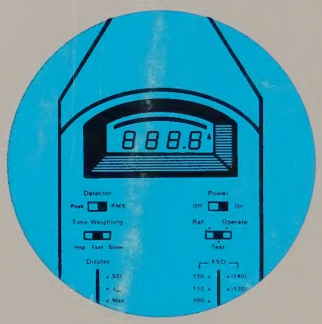
A bit may be set in the Special Event Register, for example, when a key is pressed. In that case bit 1 is set. If bit 1 is also set in the Special Event Enable Register then the Special Event bit, which is bit 8, of the Status Byte is set.

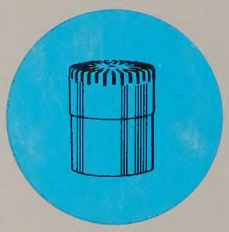


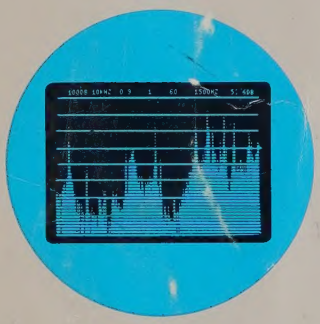


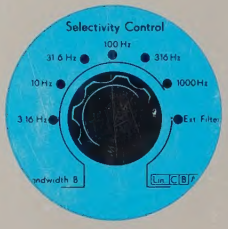


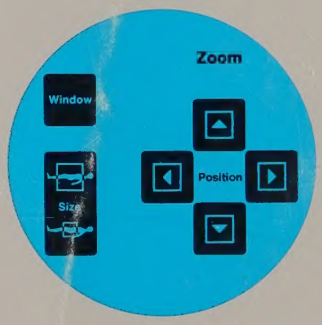


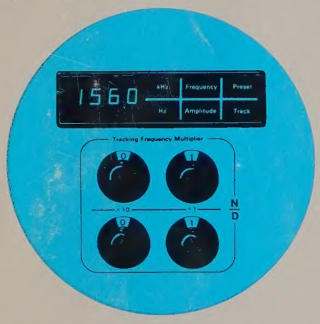


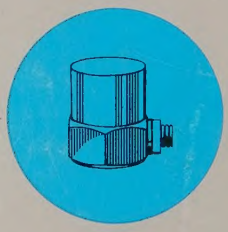


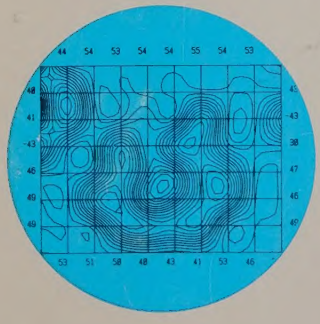


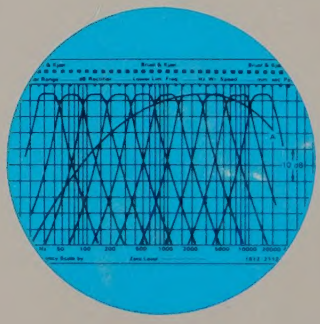












Brüel & Kjær

DK-2850 Nærum · Denmark · Telephone: +45 42 80 05 00 · Telex: 37316 bruka dk · Fax: +45 42 80 14 05

PRINTED IN DENMARK BY NÆRUM OFFSET